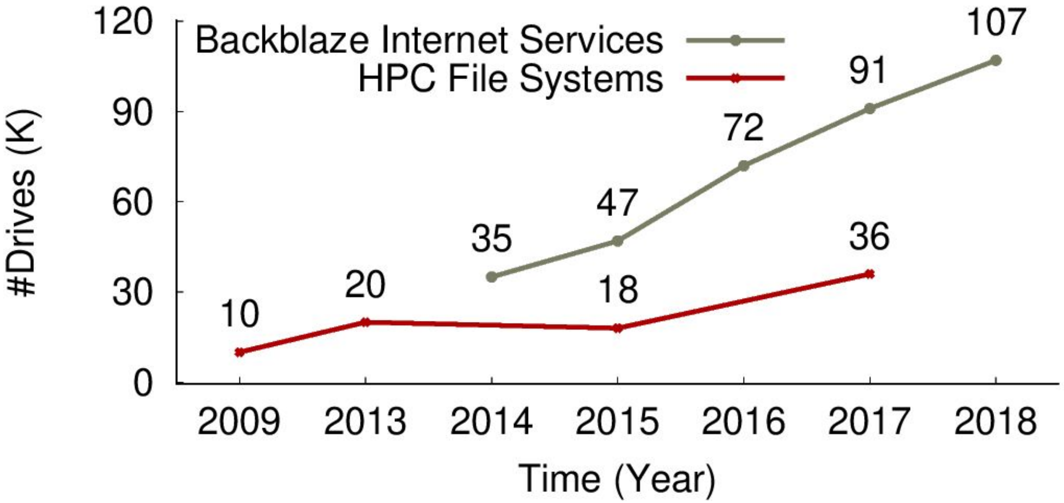
Surviving a Disk Apocalypse with Single-Overlap Declustered Parity

Huan Ke,[1] Brad Settlemyer,[2] David Bonnie,[2]
Dominic Manno,[2] John Bent,[3] Haryadi S. Gunawi[1]
[1]The University of Chicago, [2]Los Alamos National Laboratory,
[3]Seagate Technology

Increasing Disk Drives

Storage systems composing of tens of thousands of disks are increasingly common, and failure bursts become a critical concern.

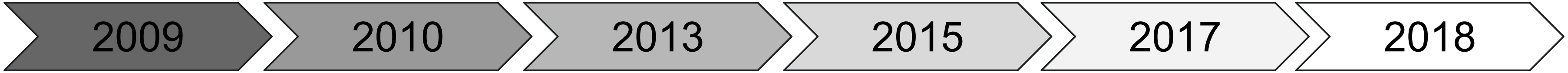


Failure Bursts

"A large fraction of failures happens in bursts"- Google, OSDI'10

"It's possible for multiple disks in the same RAID to fail simultaneously"- RAIDShield, Fast'15

"Disk failures are very common due to the large number of inexpensive disks."- OI-RAID, TPDS'18



"We present a compromise solution that use multi-level redundancy coding to reduce the probability of data loss from multiple simultaneous device failures." - MASCOTS'09

"large-scale correlated failures such as cluster power outages, a common type of data center failure scenario, are handled poorly by random replication"- Copysets, ATC'13

"To protect customer data against catastrophic data center failures, Microsoft Azure Storage optionally replicates data to a secondary DCs hundreds of miles away" - Giza, ATC'17

Empirical Failures

PlanetLab (450 nodes)

- ❑ Experience more than 35 failures within a few minutes.

Google (1000-7000 nodes)

- ❑ Large failure bursts containing 10~300 failures.

Facebook (3000 nodes)

- ❑ Up to 110 failures per day

LANL (8000-18000 disks)

- ❑ 432 disk failures within 24 hours (MarFS)
- ❑ Within a single enclosure 5 drive failures in less than 5 days (Trinity)

"… would like to optimize for minimizing the probability of incurring any data loss."

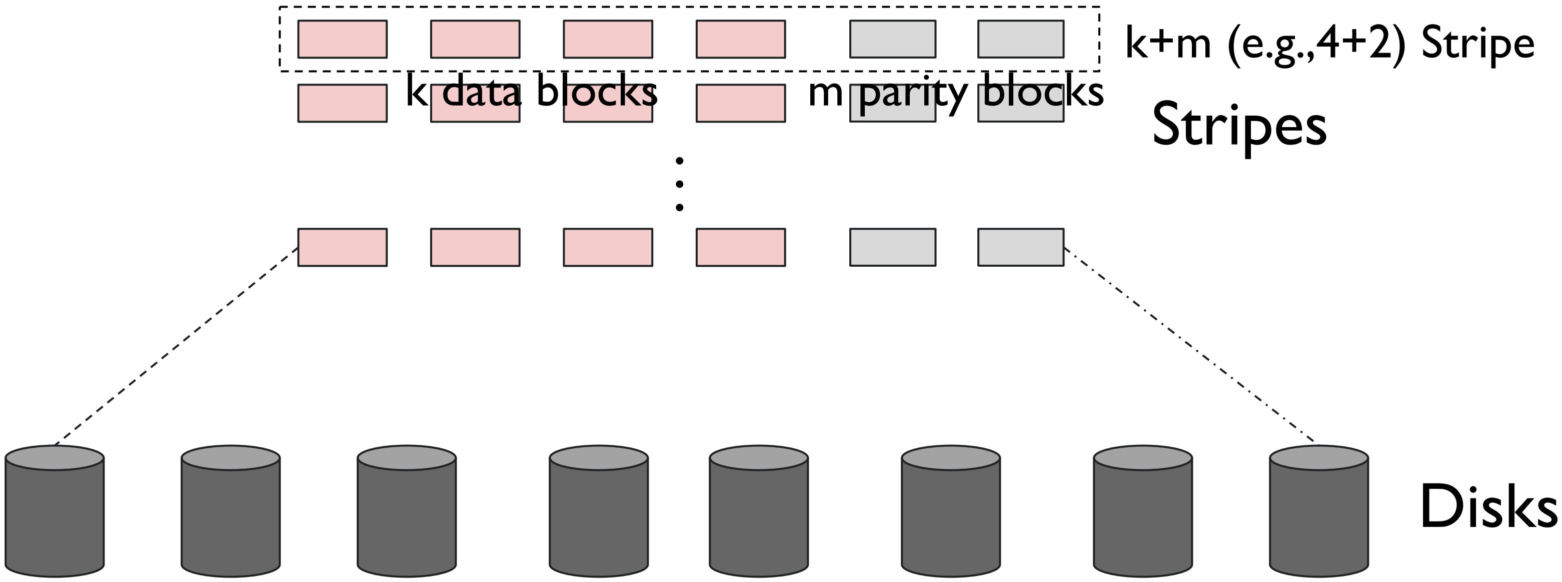
- **Stanford Researchers**

Copysets: Reducing the Frequency of Data Loss in Cloud Storage [ATC'13]

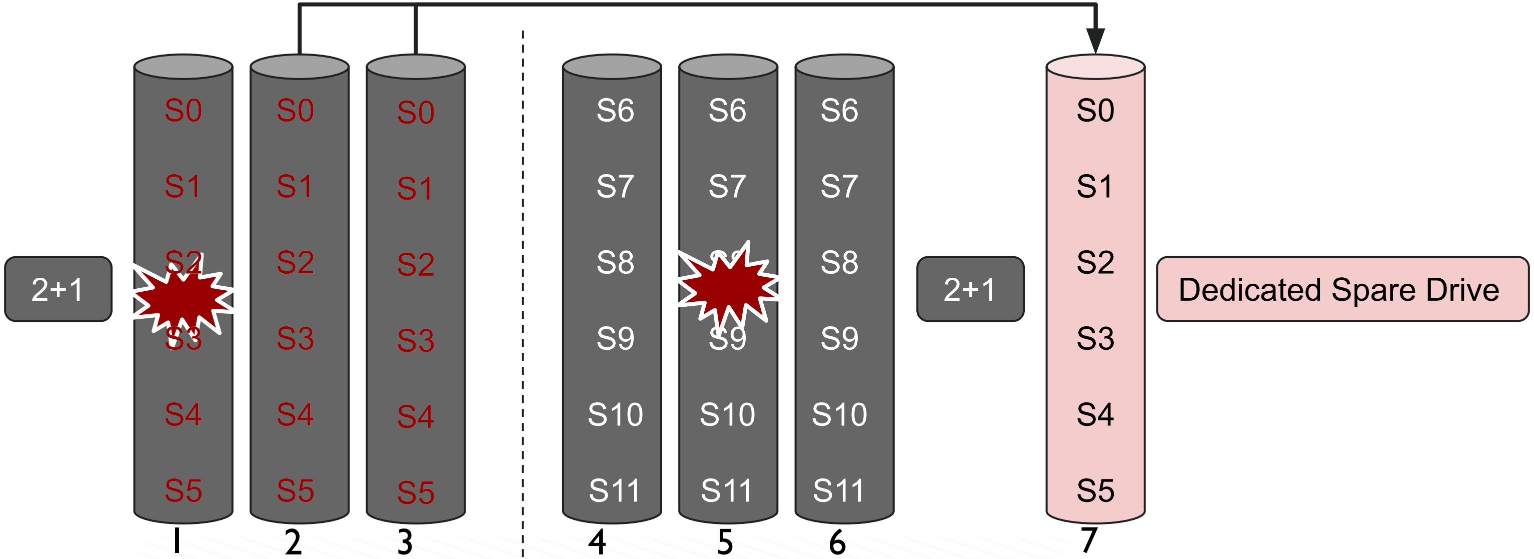
Minimizing data loss probability in the presence of failure bursts is **important**!

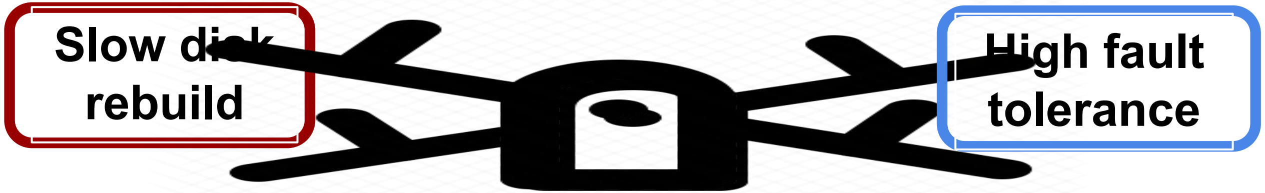
Erasure Codes

Each stripe is independently encoded and distributed across disks.

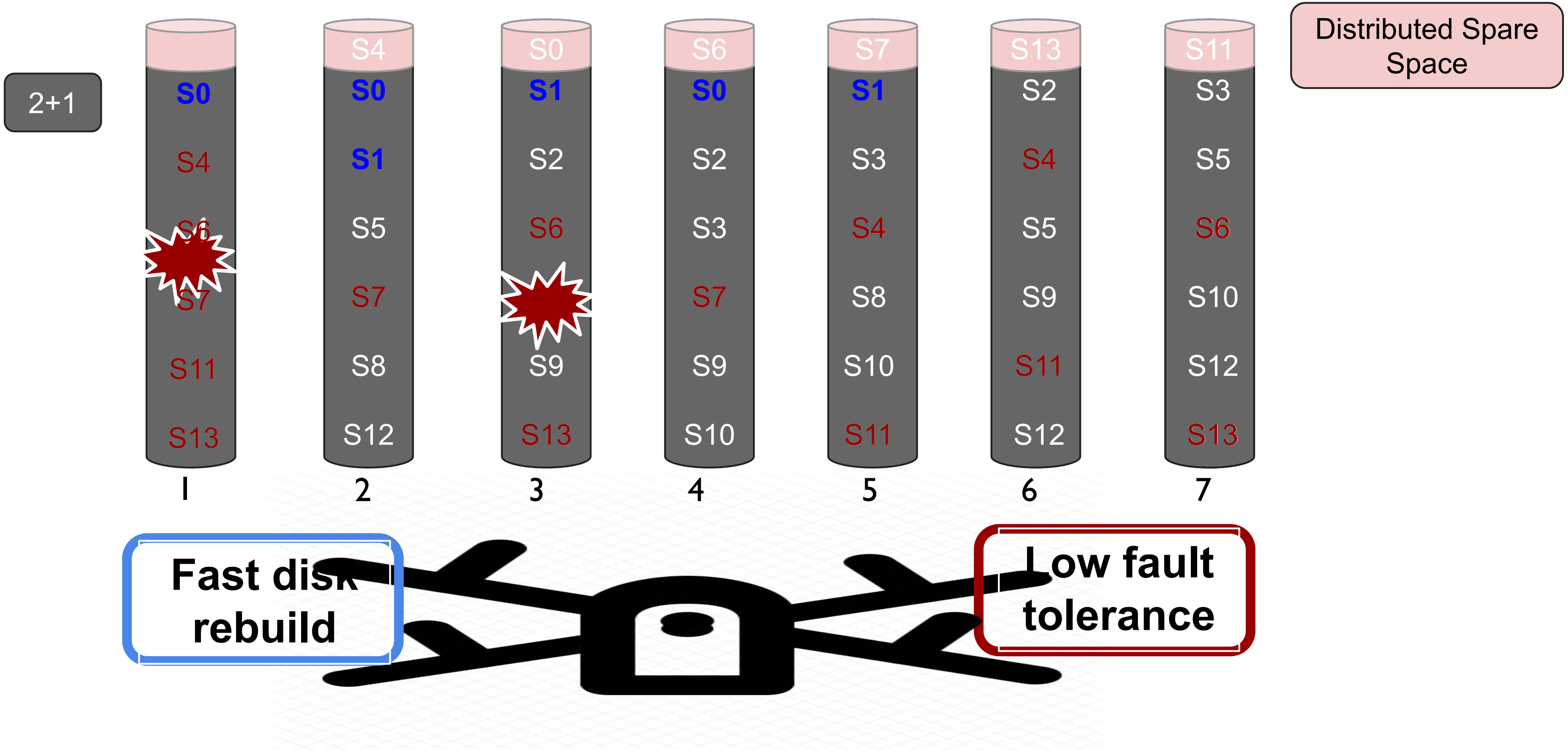


Traditional RAID





Declustered Parity



2+1

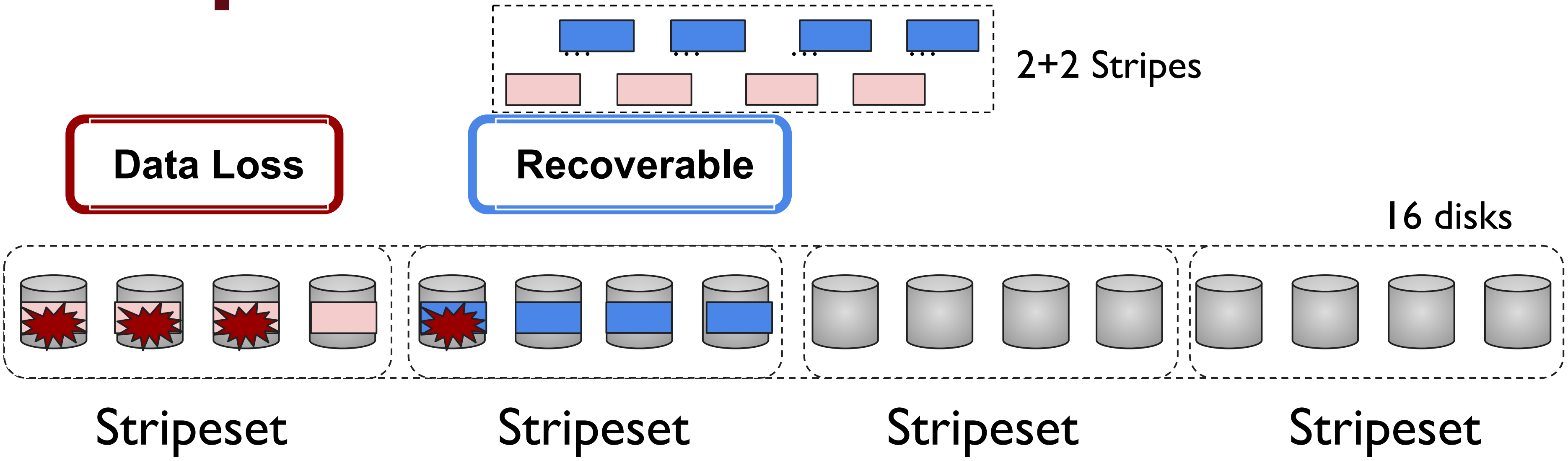
Distributed Spare Space

1	2	3	4	5	6	7
	S4	S0	S6	S7	S13	S11
S0	S0	S1	S0	S1	S2	S3
S4	S1	S2	S2	S3	S4	S5
S6	S5	S6	S3	S4	S5	S6
S7	S7		S7	S8	S9	S10
S11	S8	S9	S9	S10	S11	S12
S13	S12	S13	S10	S11	S12	S13

Fast disk rebuild

Low fault tolerance

Stripeset

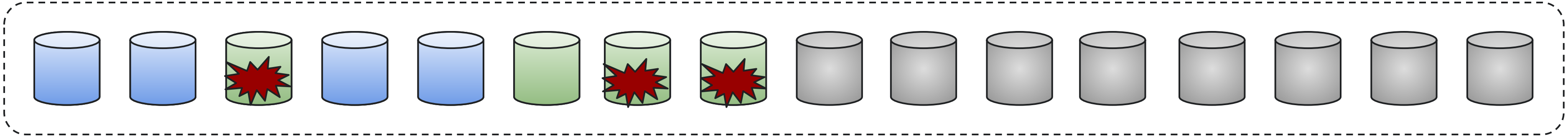


Stripeset → a set of disks for placing a stripe

Probability of Data Loss (PDL)

Maximal PDL 100%

2+2



$\binom{16}{4}$ 1820 Stripesets

The 3 failures will definitely be within a stripeset.

Single-Overlap Stripesets

Each pair of two disks appears in a single stripeset.

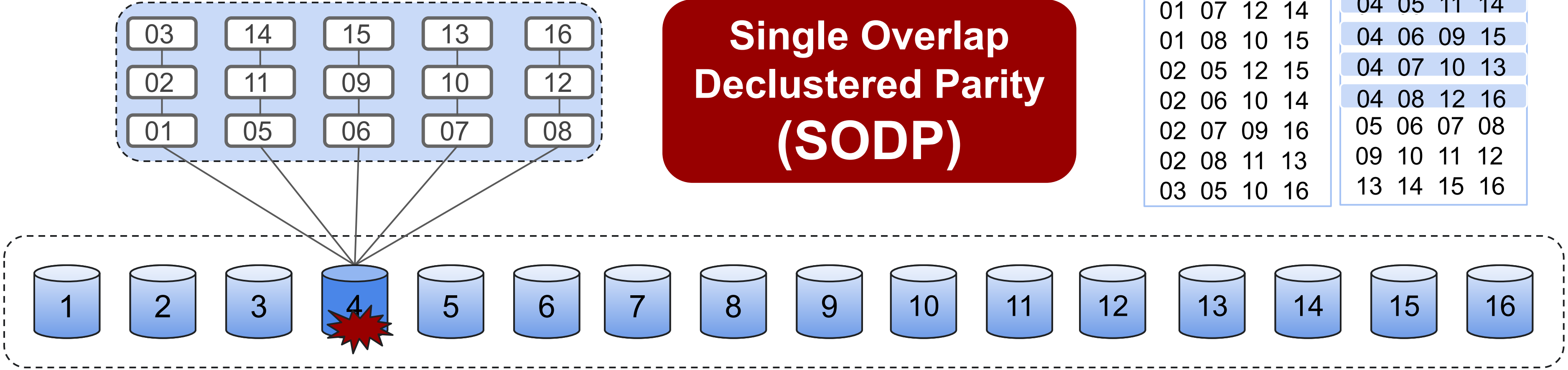
- ❑ Declustered data layout with minimal stripesets
- ❑ At most one overlap between any two stripesets

Single Overlap Stripesets

01 02 03 04
01 05 09 13
01 06 11 16
01 07 12 14
01 08 10 15
02 05 12 15
02 06 10 14
02 07 09 16
02 08 11 13
03 05 10 16

03 06 12 13
03 07 11 15
03 08 09 14
04 05 11 14
04 06 09 15
04 07 10 13
04 08 12 16
05 06 07 08
09 10 11 12
13 14 15 16

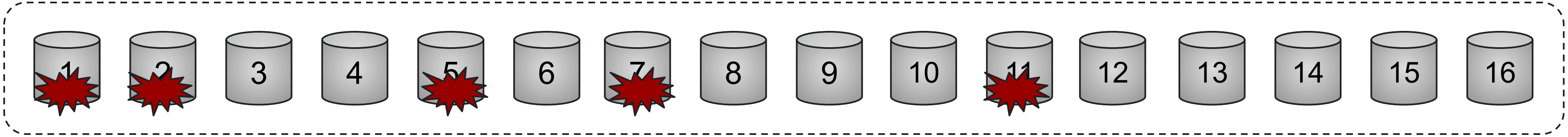
Single Overlap Declustered Parity (SODP)



Multiple Disk Failures

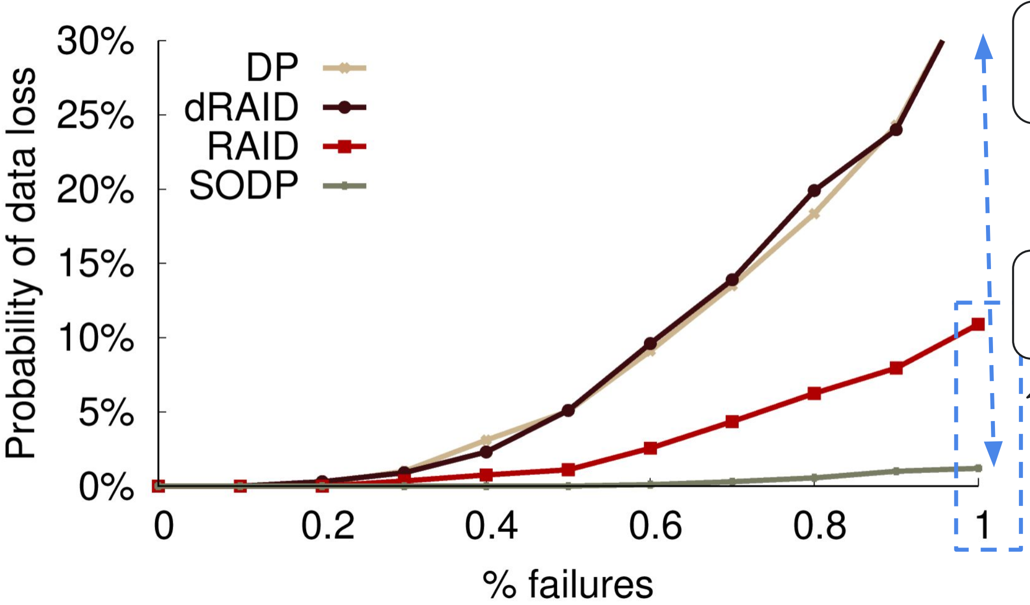
Each affected stripeset just has two disk failures.

SODP							
01	02	03	04	03	06	12	13
01	05	09	13	03	07	11	15
01	06	11	16	03	08	09	14
01	07	12	14	04	05	11	14
01	08	10	15	04	06	09	15
02	05	12	15	04	07	10	13
02	06	10	14	04	08	12	16
02	07	09	16	05	06	07	08
02	08	11	13	09	10	11	12
03	05	10	16	13	14	15	16



SODP Evaluation

Probability of data loss under a burst of failures within 24h.



PDL 30x reduced compared to DP

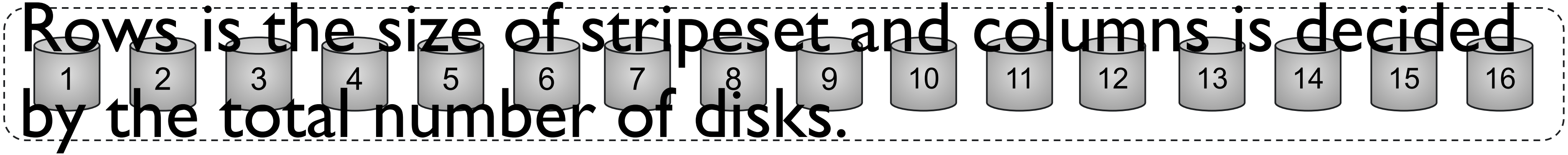
PDL 10x reduced compared to RAID.

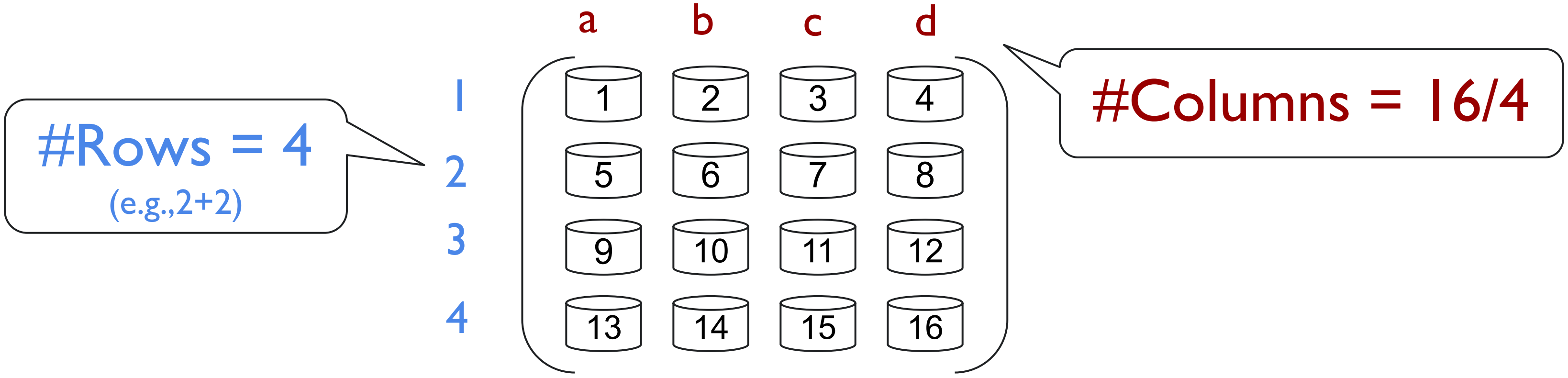
SODP Algorithm

- Column-based stripesets
- Row-column stripesets
- Row-based stripesets

SODP Algorithm

Rows is the size of stripeset and columns is decided by the total number of disks.





#Rows = 4
(e.g.,2+2)

$$\begin{array}{c|cccc} & a & b & c & d \\ \hline 1 & 1 & 2 & 3 & 4 \\ 2 & 5 & 6 & 7 & 8 \\ 3 & 9 & 10 & 11 & 12 \\ 4 & 13 & 14 & 15 & 16 \end{array}$$

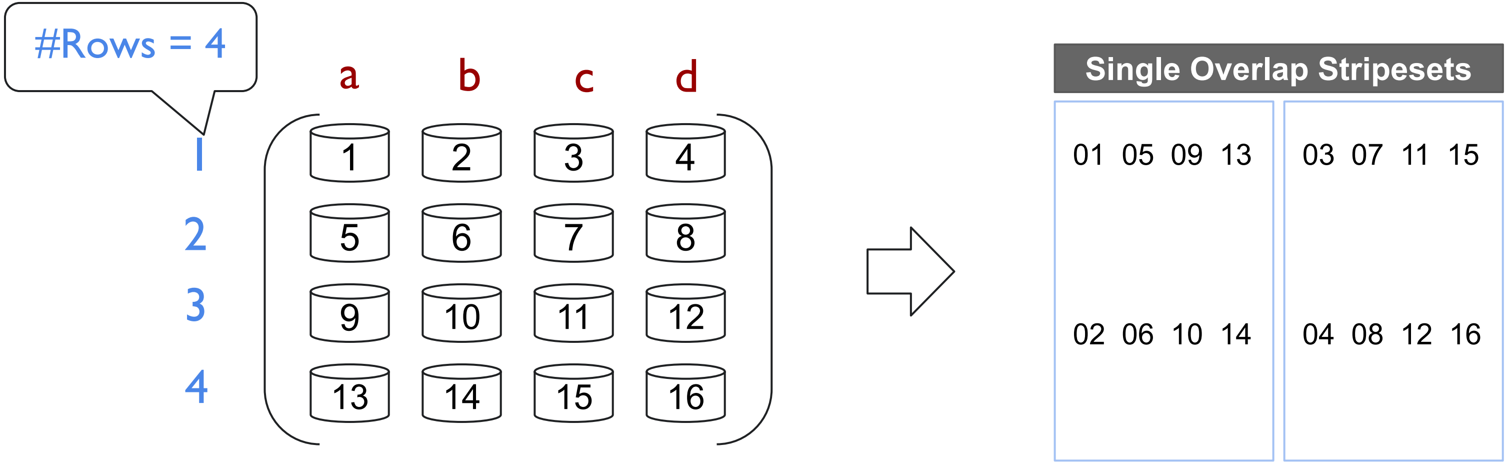
#Columns = 16/4

Disk Matrix

SODP Algorithm

Column-based stripesets

❑ Since the number of rows is equal to the size of one stripeset, then each column consists of a stripeset.



#Rows = 4

	a	b	c	d
1	1	2	3	4
2	5	6	7	8
3	9	10	11	12
4	13	14	15	16

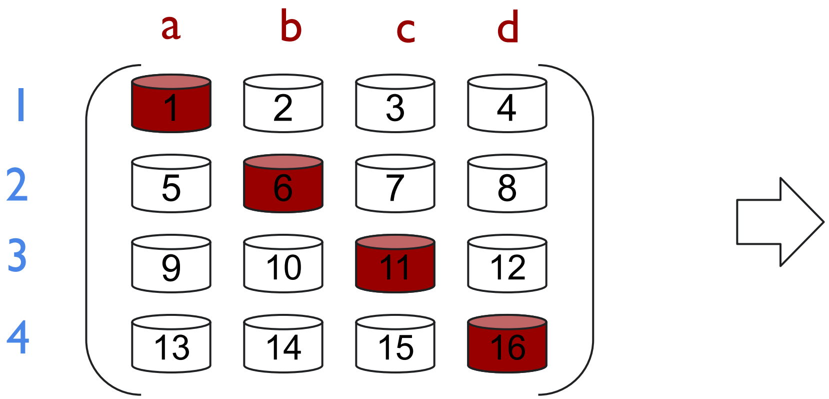
Single Overlap Stripesets

01 05 09 13 02 06 10 14	03 07 11 15 04 08 12 16

SODP Algorithm

Row-column stripesets

- Choose disks from different rows and different columns (e.g., diagonal disks)



	a	b	c	d
1	1	2	3	4
2	5	6	7	8
3	9	10	11	12
4	13	14	15	16

Single Overlap Stripesets

01 05 09 13
01 06 11 16

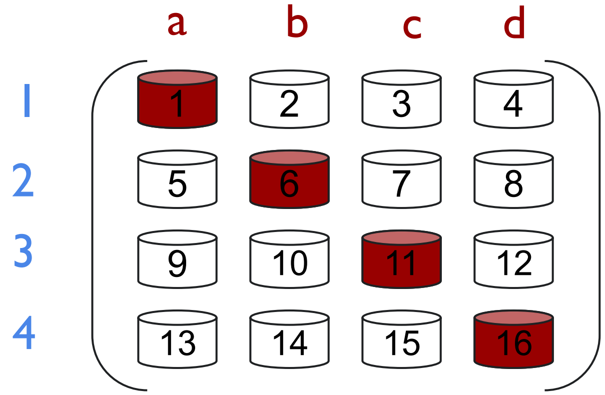
02 06 10 14

03 07 11 15

04 08 12 16

SODP Algorithm

Column-relative position array for stripeset by using disks from different rows and columns.



	a	b	c	d
1	**1**	2	3	4
2	5	**6**	7	8
3	9	10	**11**	12
4	13	14	15	**16**

rowId columnId

[(1 a), (2 b),(3 c), (4 d)]

⇩

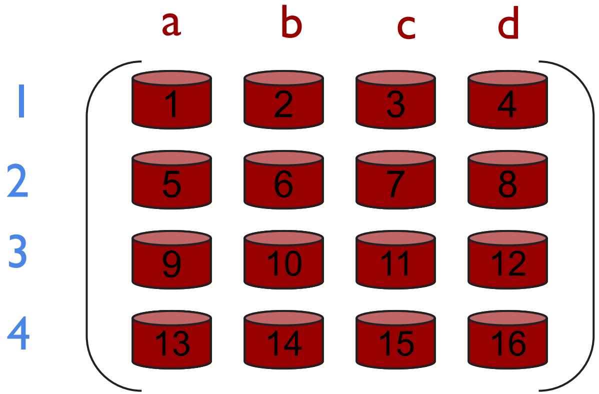
a b c d

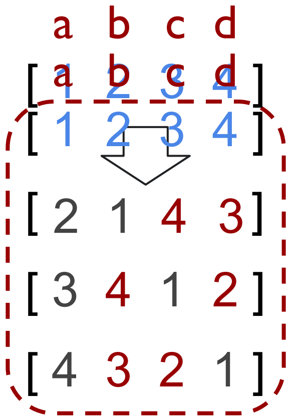
[1, 2, 3, 4]

position array

SODP Algorithm

Permutation shuffle by swapping any pair of two positions to generate 4 non-overlapped arrays.





non-overlap

Single Overlap Stripesets

01 05 09 13
01 06 11 16

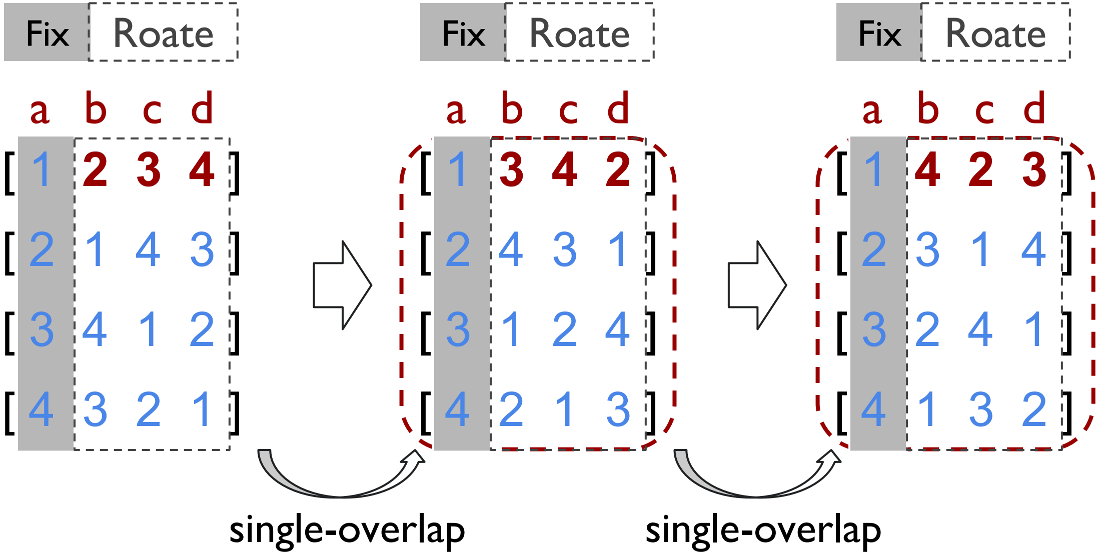
02 06 10 14
05 02 15 12

03 07 11 15
09 14 03 08

04 08 12 16
13 10 07 04

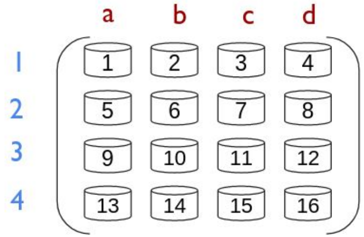
SODP Algorithm

Fix one position and rotate other positions in arrays.



Fix | Roate

$$\begin{array}{cccc} a & b & c & d \\ {[}1 & 2 & 3 & 4{]} \\ {[}2 & 1 & 4 & 3{]} \\ {[}3 & 4 & 1 & 2{]} \\ {[}4 & 3 & 2 & 1{]} \end{array} \Rightarrow \begin{array}{cccc} a & b & c & d \\ {[}1 & 3 & 4 & 2{]} \\ {[}2 & 4 & 3 & 1{]} \\ {[}3 & 1 & 2 & 4{]} \\ {[}4 & 2 & 1 & 3{]} \end{array} \Rightarrow \begin{array}{cccc} a & b & c & d \\ {[}1 & 4 & 2 & 3{]} \\ {[}2 & 3 & 1 & 4{]} \\ {[}3 & 2 & 4 & 1{]} \\ {[}4 & 1 & 3 & 2{]} \end{array}$$



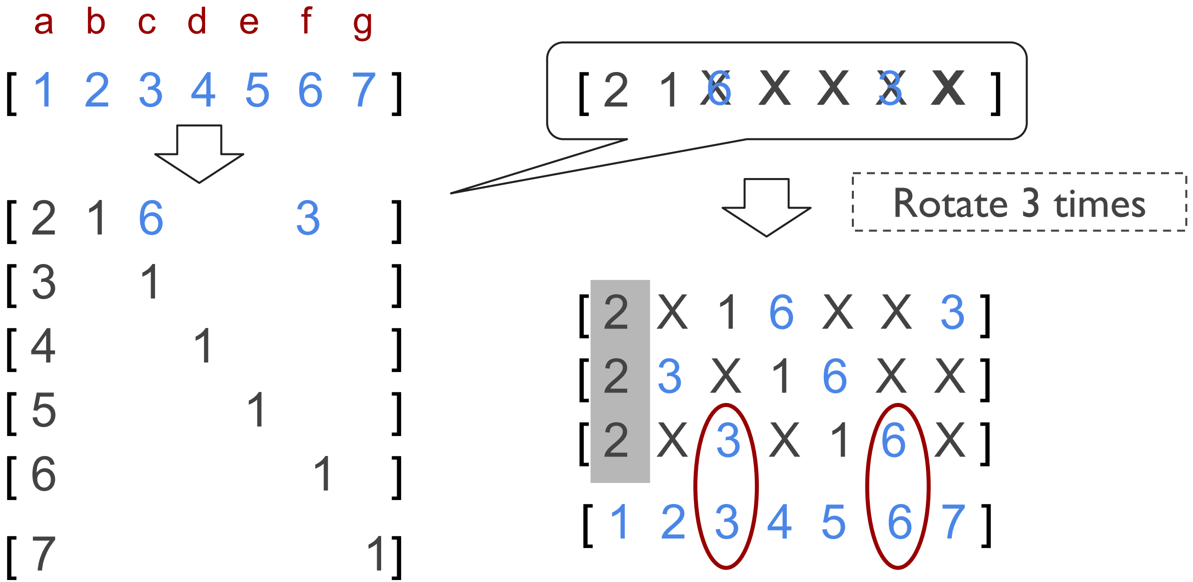
$$\begin{array}{c|cccc} & a & b & c & d \\ \hline 1 & 1 & 2 & 3 & 4 \\ 2 & 5 & 6 & 7 & 8 \\ 3 & 9 & 10 & 11 & 12 \\ 4 & 13 & 14 & 15 & 16 \end{array}$$

Single Overlap Stripesets

01 05 09 13	03 07 11 15
01 06 11 16	09 14 03 08
01 10 15 08	09 02 07 16
01 14 07 12	09 06 15 04
02 06 10 14	04 08 12 16
05 02 15 12	13 10 07 04
05 14 11 04	13 06 03 12
05 10 03 16	13 02 11 08

SODP Algorithm

Swap permutation and rotation work for all?



$$\begin{array}{c|ccccccc|}
 & a & b & c & d & e & f & g \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7
\end{array}$$

$$\begin{bmatrix}
2 & 1 & 6 & & & 3 & \\
3 & & 1 & & & & \\
4 & & & 1 & & & \\
5 & & & & 1 & & \\
6 & & & & & 1 & \\
7 & & & & & & 1
\end{bmatrix}$$

[2 1 ~~X~~6 X X ~~X~~3 X]

Rotate 3 times

[2 X 1 6 X X 3]

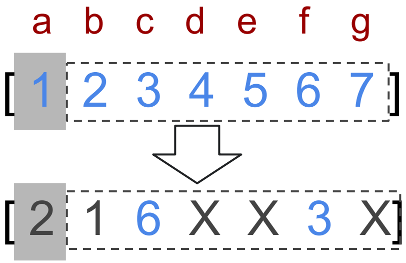
[2 3 X 1 6 X X]

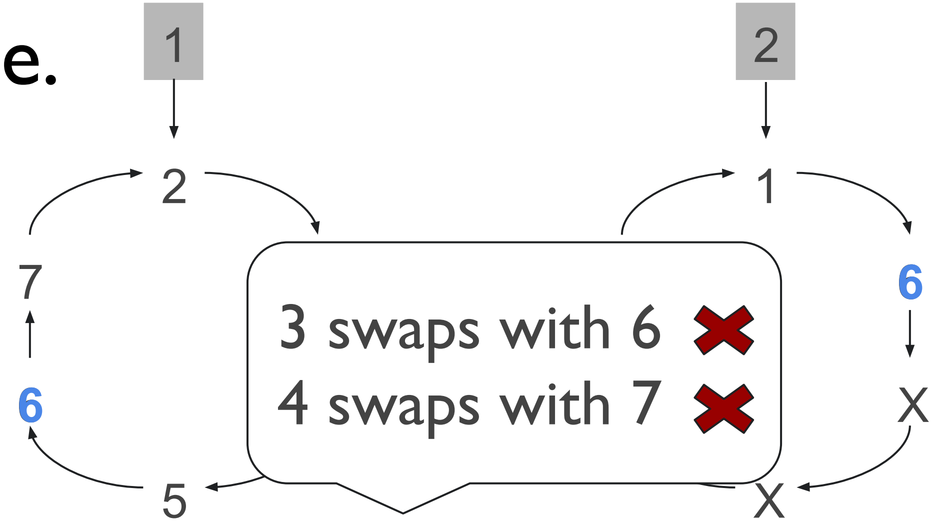
[2 X 3 X 1 6 X]

[1 2 3 4 5 6 7]

SODP Algorithm

Rotate distance represents the distance between positions in the circle.





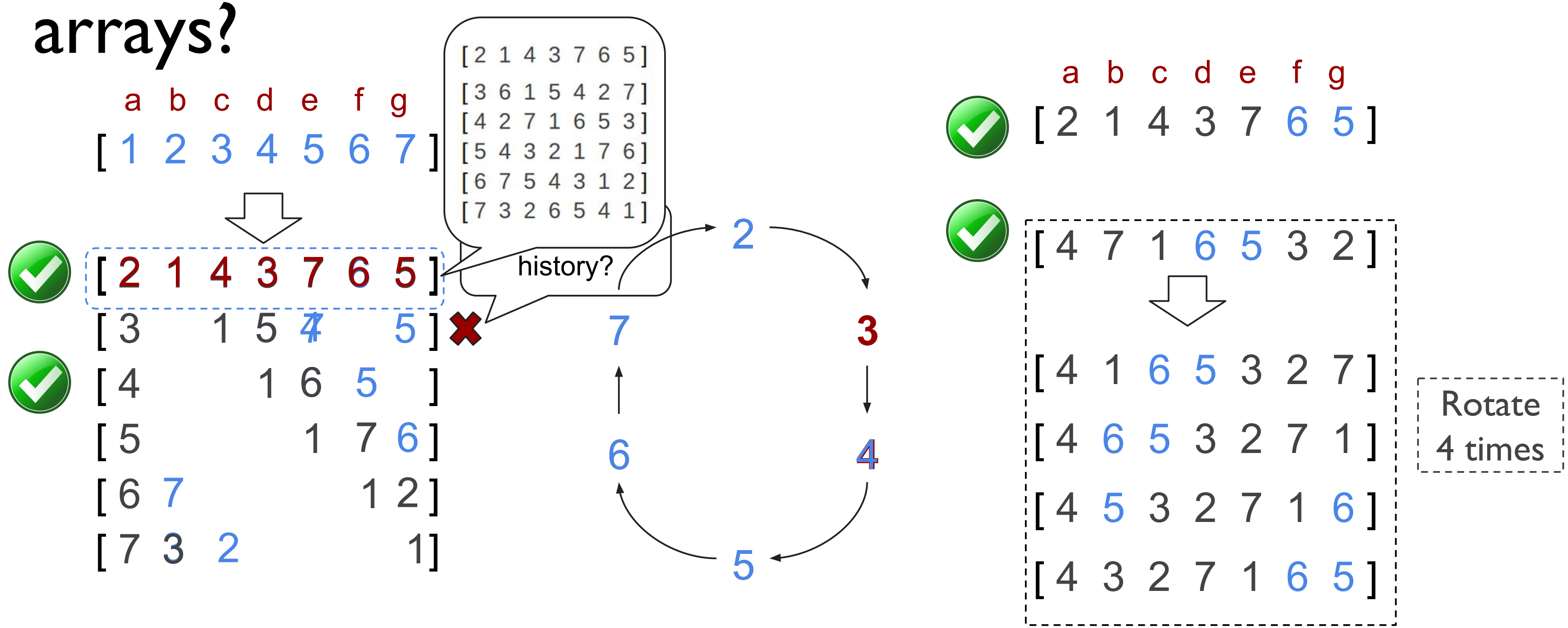
3 swaps with 6 ✗

4 swaps with 7 ✗

Constraint 1: Rotate distance before and after swapping can not be equal. ⇒ avoid multiple overlaps with the diagonal array

SODP Algorithm

How to avoid multiple overlaps with other position arrays?



a	b	c	d	e	f	g
1	2	3	4	5	6	7
2	1	4	3	7	6	5
3		1	5	4		5
4			1	6	5	
5				1	7	6
6	7				1	2
7	3	2				1

```
[2 1 4 3 7 6 5]
[3 6 1 5 4 2 7]
[4 2 7 1 6 5 3]
[5 4 3 2 1 7 6]
[6 7 5 4 3 1 2]
[7 3 2 6 5 4 1]
```

history?

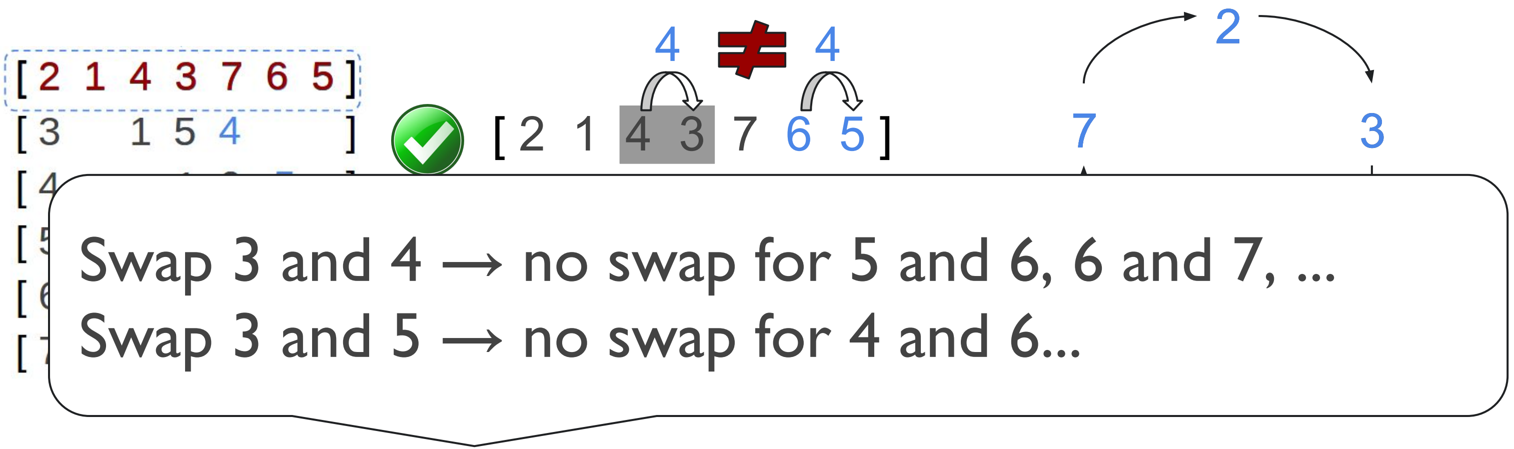
a	b	c	d	e	f	g
2	1	4	3	7	6	5

```
[4 7 1 6 5 3 2]

[4 1 6 5 3 2 7]
[4 6 5 3 2 7 1]
[4 5 3 2 7 1 6]
[4 3 2 7 1 6 5]
```

Rotate 4 times

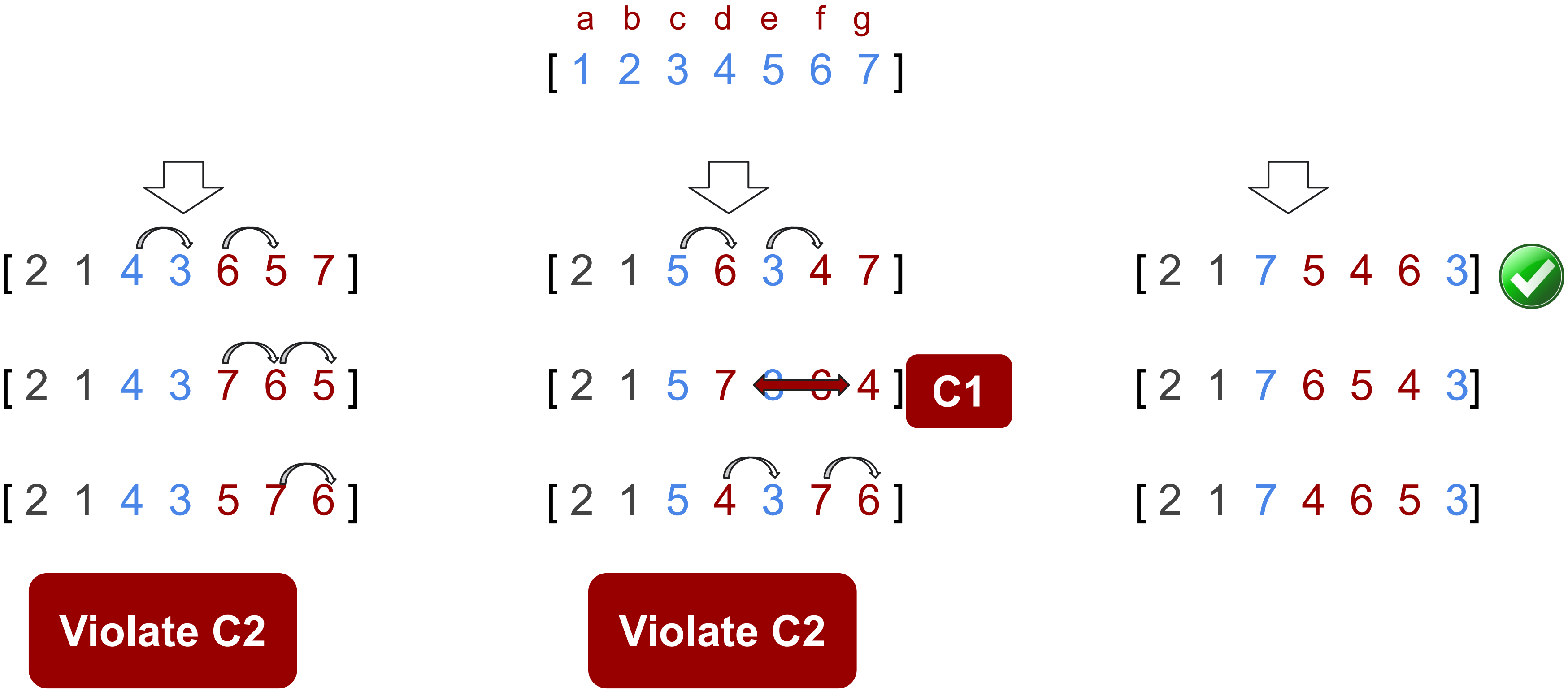
SODP Algorithm



Swap 3 and 4 ⟶ no swap for 5 and 6, 6 and 7, ...
Swap 3 and 5 ⟶ no swap for 4 and 6...

Constraint II: Distinct rotate distances in the second position array. ⇒ avoid multiple overlaps with the other position arrays

SODP Algorithm



a b c d e f g

[1 2 3 4 5 6 7]

[2 1 4 3 6 5 7]

[2 1 4 3 7 6 5]

[2 1 4 3 5 7 6]

Violate C2

[2 1 5 6 3 4 7]

[2 1 5 7 3 6 4] C1

[2 1 5 4 3 7 6]

Violate C2

[2 1 7 5 4 6 3] ✓

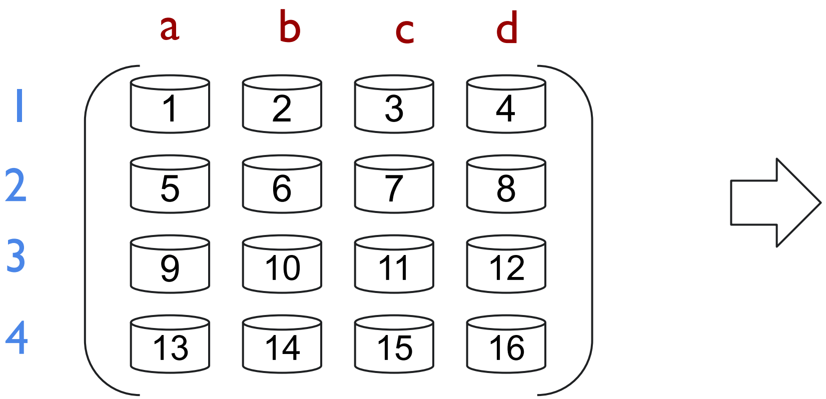
[2 1 7 6 5 4 3]

[2 1 7 4 6 5 3]

SODP Algorithm

Row-based stripesets

- ❑ Each row consists of 4 disks, which is exactly one stripeset



	a	b	c	d
1	1	2	3	4
2	5	6	7	8
3	9	10	11	12
4	13	14	15	16

Single Overlap Stripesets

01	05	09	13	03	07	11	15
01	02	03	04	09	10	11	12
01	06	11	16	09	14	03	08
01	10	15	08	09	02	07	16
01	14	07	12	09	06	15	04
02	06	10	14	04	08	12	16
05	02	15	12	13	10	07	04
05	14	11	04	13	06	03	12
05	10	03	06	13	02	11	08
05	06	07	08	13	14	15	16

- ❑ If each row consists more than 4 disks, it will lead to single overlap with some disks but zero-overlap with other disks, please follow our next talk FODP.

SODP Algorithm

- Column-based stripesets
- Row-column stripesets
  - Permutation shuffle with swaps and rotation
  - Constraint I to single overlap with diagonal array
  - Constraint II to single overlap with derived arrays
- Row-based stripesets

SODP Conclusion

"Why should we address failure bursts?"

Storage systems scaling out!

Failures bursts are **common**!

❑ Highlight overlooked fault tolerance of declustered parity and guarantee the identical rebuild.

❑ Fractional Overlap Declustered Parity (FODP) next!

CERES

Thank you!
Questions?

THE UNIVERSITY OF CHICAGO

Los Alamos NATIONAL LABORATORY EST. 1943

Seagate

http://ucare.cs.uchicago.edu