Self-Adjusting Two-Failure Tolerant Disk Arrays

Ignacio Corderí, Universidad Católica del Uruguay
Thomas Schwarz,S.J., Universidad Católica del Uruguay
Ahmed Amer, Santa Clara University
Darrell D. E. Long, UC Santa Cruz
J.F. Pâris, University of Houston

Self-Adjusting Two-Failure Tolerant Disk Arrays

- Disk arrays suffer from disk failures:
  - Reliability in large scale "real" storage facilities is surprisingly high
    - 1.7% - 8.6% Annual Failure Rate (AFR) observed by Pinheiro et al.
    - 0.5% - 13.5% AFR observed by Schroeder and Gibson
- Many disks develop latent sector failures:
  - Data is lost on a single or a few sectors
    - 3.45% over 32 months according to Bairavasundaram et al. 2008
- Disks are not the only failure mechanism:
  - Disk Failure (20%-55% in study by Jiang et al, 2008)
  - Physical Interconnect Failure (27%–68% in the same study)
  - Protocol Failure and performance failure are also important

- E. Pinheiro, W. Weber, and L. Barroso, "Failure trends in a large disk drive population," FAST, 2007.
- B. Schroeder and G. Gibson, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?", FAST, 2007
- L. Bairavasundaram, G. Goodson, S. Pasupathy, and J. Schindler, "An analysis of latent sector errors in disk drives," SIGMETRICS 2008
- L. Bairavasundaram, A. Arpaci-Dusseau, R. Arpaci-Dusseau, G. Goodson, and B. Schroeder: "An analysis of data corruption in the storage stack" ACM Transactions on Storage (TOS), 2008
- W. Jiang, C. Hu, Y. Zhou, and A. Kanevsky: Are disks the dominant contributor for storage failures? A comprehencsive sutdy of storage subsystem failure characteristics, ACM Transactions on Storage (TOS), 2008

Self-Adjusting Two-Failure Tolerant Disk Arrays

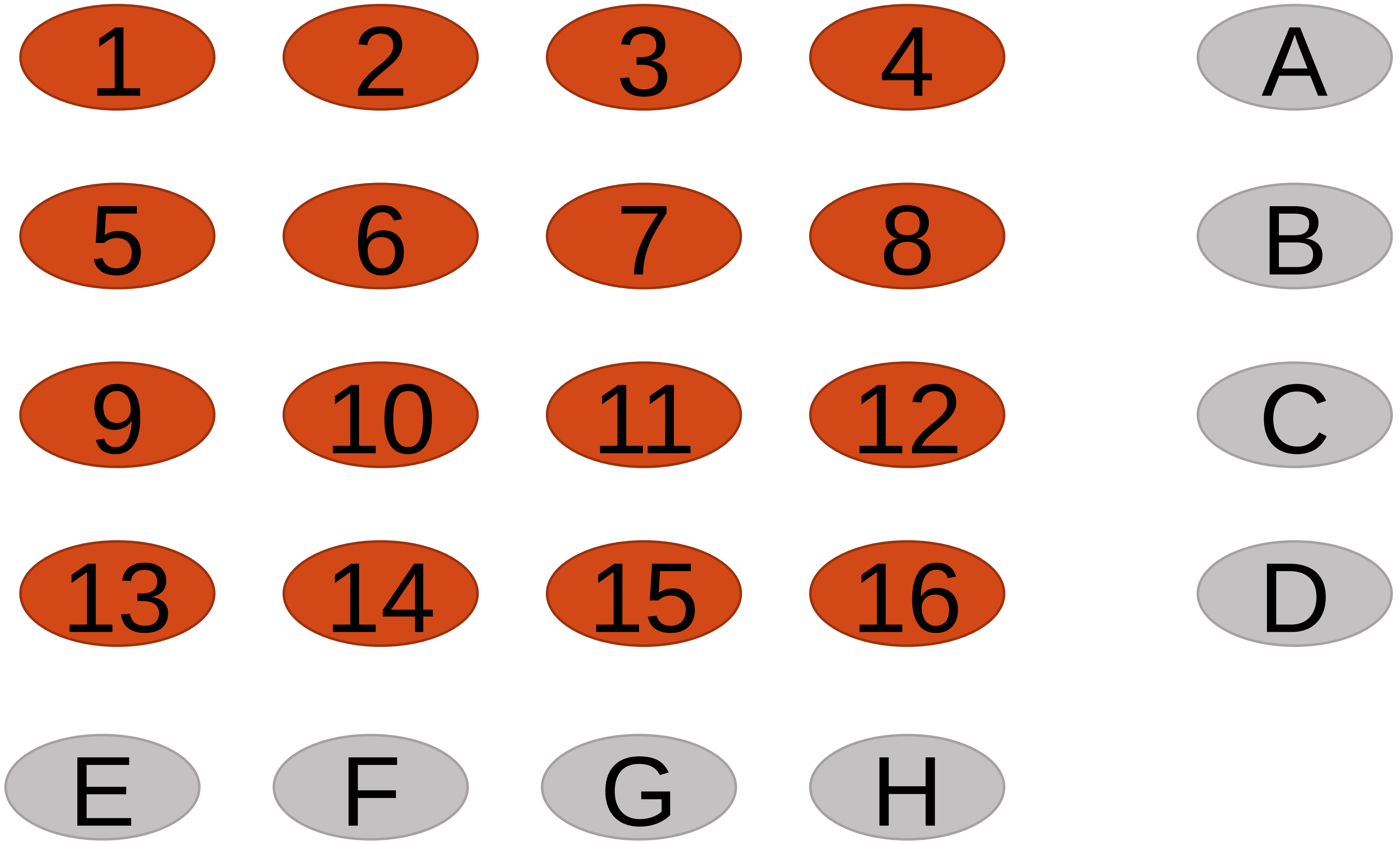
- As we know, to protect user data, it has to be stored redundantly
  - Mirroring / Replication
    - Same data is stored twice / several times
    - Good performance, good reliability, high storage overhead
  - Parity / Erasure coding
    - Bad to reasonable performance
      - Can be alleviated by caching, large writes, …
    - Good reliability
    - Low storage overhead

Self-Adjusting Two-Failure Tolerant Disk Arrays

- 2d-layout (Hellerstein, et al)
  - Places each data disk in two parity blocks
  - Uses a square layout
- General layout:
  - Data is stored in disklets (of fixed size)
  - A number of disklets is stored at a single disk
    - Allows use of different types of disks
  - Layout: Each data disklet is in exactly two parity stripes
    - Higher failure tolerance is usually not needed
    - Higher failure tolerance costs in storage and performance

Coding techniques for handling failures in large disk arrays
L.Hellerstein, GA Gibson, RM Karp, RH Katz, DA - Algorithmica, 1994 - Springer

2-d layout with 16 data and 8 parity disks



Self-Adjusting Two-Failure Tolerant Disk Arrays

- Criteria for good layout:
  - Each reliability stripe consists of *n* data disklets and one parity disklet
  - Each disk contains the same number of parity disklets
    - To equalize write load
  - Each disk contains the same number of data disklets
    - To equalize write and read load
  - Each disklet contains the same number of unassigned disklets
    - Spare space to be used in case of disk failure
    - To equalize write and read load
  - If one disk fails, then the reconstruction load is equally distributed
    - Reads to a failed disk are satisfied by reading from all other disks in a reliability stripe containing the failed disk
    - Piggy-backing on read load, we reconstruct loss data and write it to other disks

Self-Adjusting Two-Failure Tolerant Disk Arrays

- Key Observations:
  - Large scale storage organizations are dynamic
    - Disks enter system in batches
    - Disk capacity changes over the lifetime of the system
    - Leave it through failure and decommissioning
  - Optimal layouts only for some parameters
  - Optimal layouts do not adjust well to changes
- Conclusion:
    - By applying maxim: “The better is the enemy of the good”
  - Layouts that are close to satisfying these conditions usually suffice and can be easily adapted to changing number of disks.

Self-Adjusting Two-Failure Tolerant Disk Arrays

- We store data in disklets – virtual disks of fixed size
  - Disklets are large-sized contiguous sections of disks (~10GB – 200GB ≈ 200 – 10 disklets per disk)
  - Each data disklet is placed in two reliability stripes with one parity disklet each.
  - We can move disklets transparently to other disks
    - E.g. to reorganize the disk array after failures or when adding disks to the array

Graph Representation

- Each disklet is in two reliability stripes
- Mathematical design theory knows this as a configuration:
  - Elements (data disklets) and blocks (reliability stripe)
  - Each element is in exactly two blocks
  - Each block has *n* elements
  - Two different elements share at most one block

Design Theory

Blocks are

A = {1,2,3}

B = {1,4,5}

C = {2,4,6}

D = {3,5,6}

Disk Array Layout

Stripes are

1,2,3,A

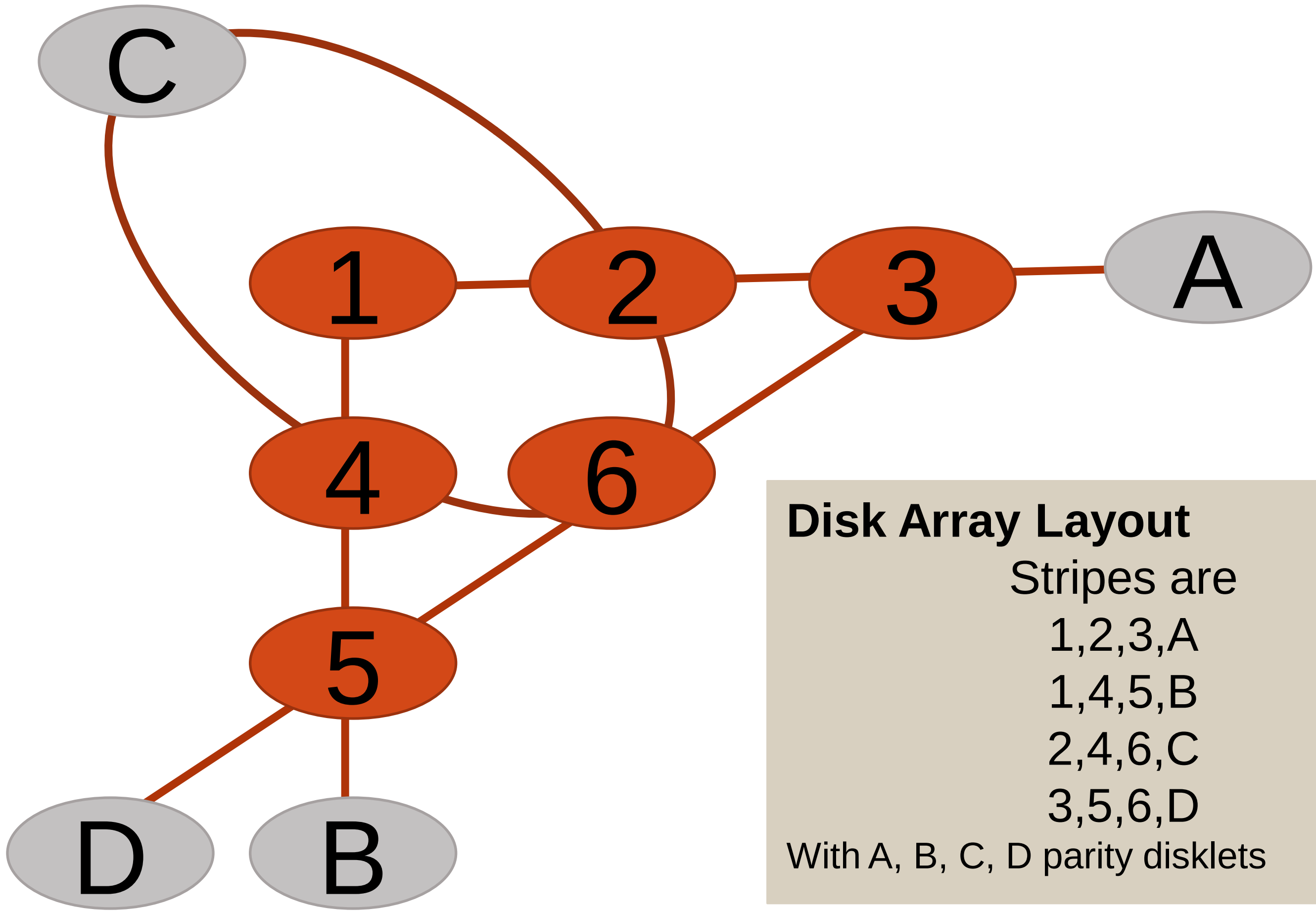
1,4,5,B

2,4,6,C

3,5,6,D

With A, B, C, D parity disklets

Graph Representation



Disk Array Layout

Stripes are

1,2,3,A

1,4,5,B

2,4,6,C

3,5,6,D

With A, B, C, D parity disklets

Graph Representation

- Dual in design theory: Blocks become elements, elements become blocks
- Dual of dual is the original design
- Dual of configuration is a regular graph.

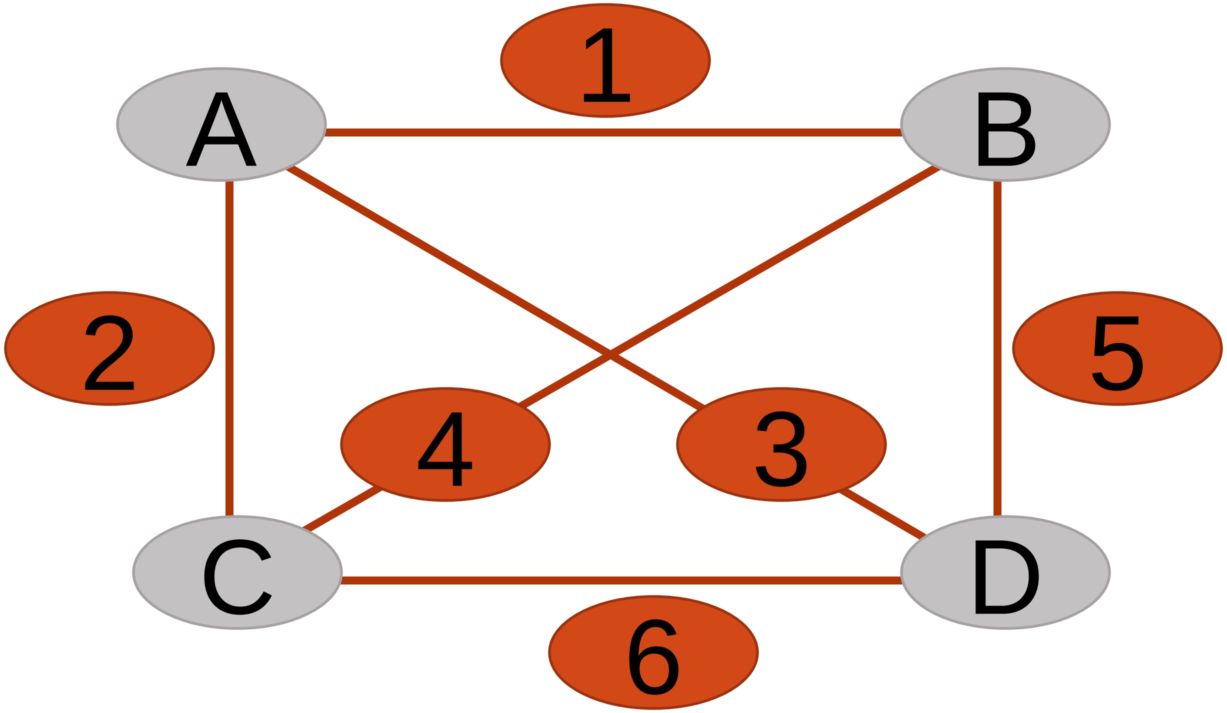
Blocks are
A = {1,2,3}
B = {1,4,5}
C = {2,4,6}
D = {3,5,6}

Stripes are
1,2,3,A
1,4,5,B
2,4,6,C
3,5,6,D
With A, B, C, D parity disklets

Dual:
1: (A,B)
2: (A,C)
3: (A,D)
4: (B,C)
5: (B,D)
6: (C,D)

Graph Representation

- Dual is a graph
  - Vertices correspond to parity
  - Edges to data



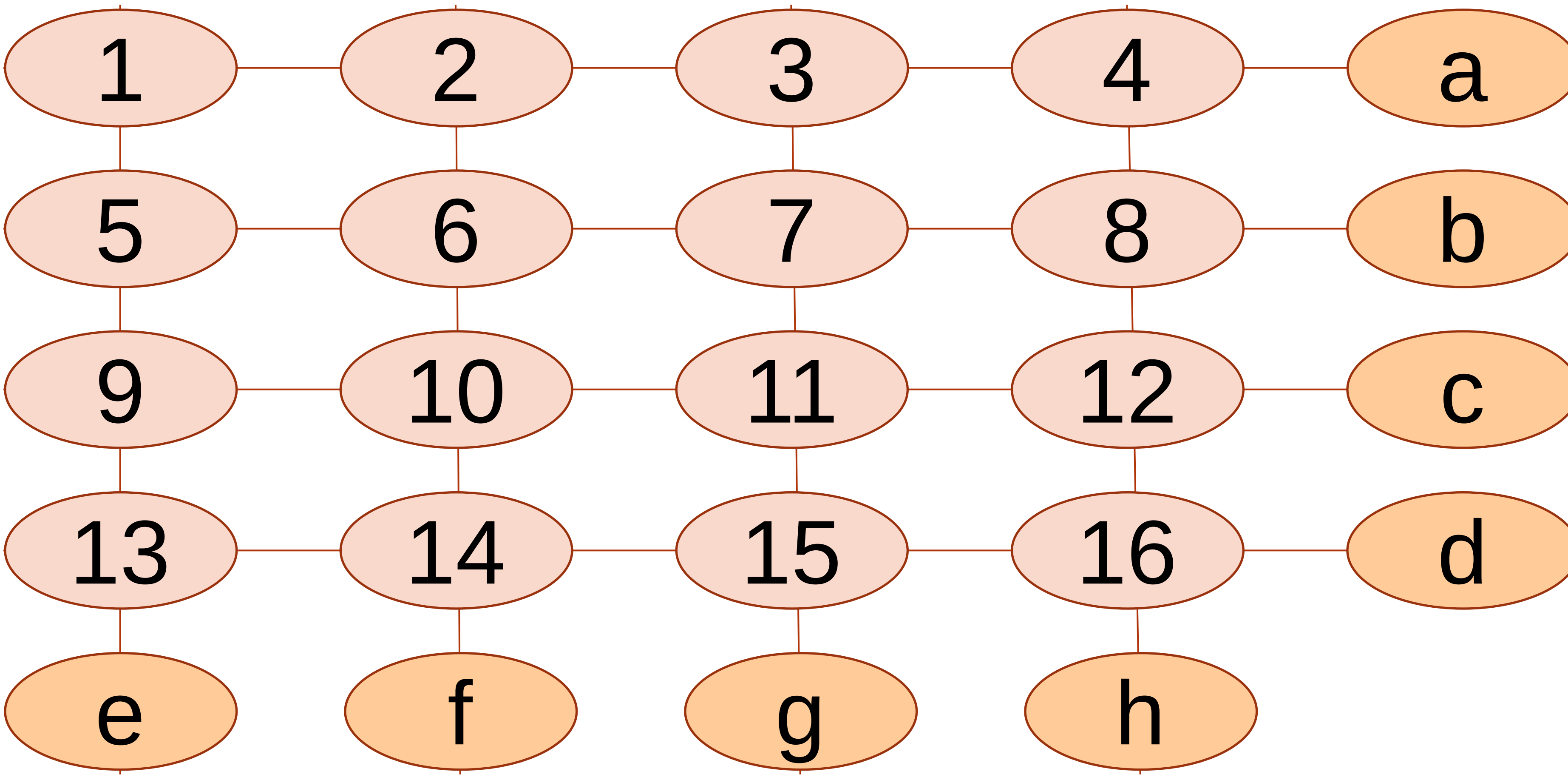
Dual:
1: (A,B)
2: (A,C)
3: (A,D)
4: (B,C)
5: (B,D)
6: (C,D)

Stripes are
1,2,3,A
1,4,5,B
2,4,6,C
3,5,6,D
With A, B, C, D parity disklets

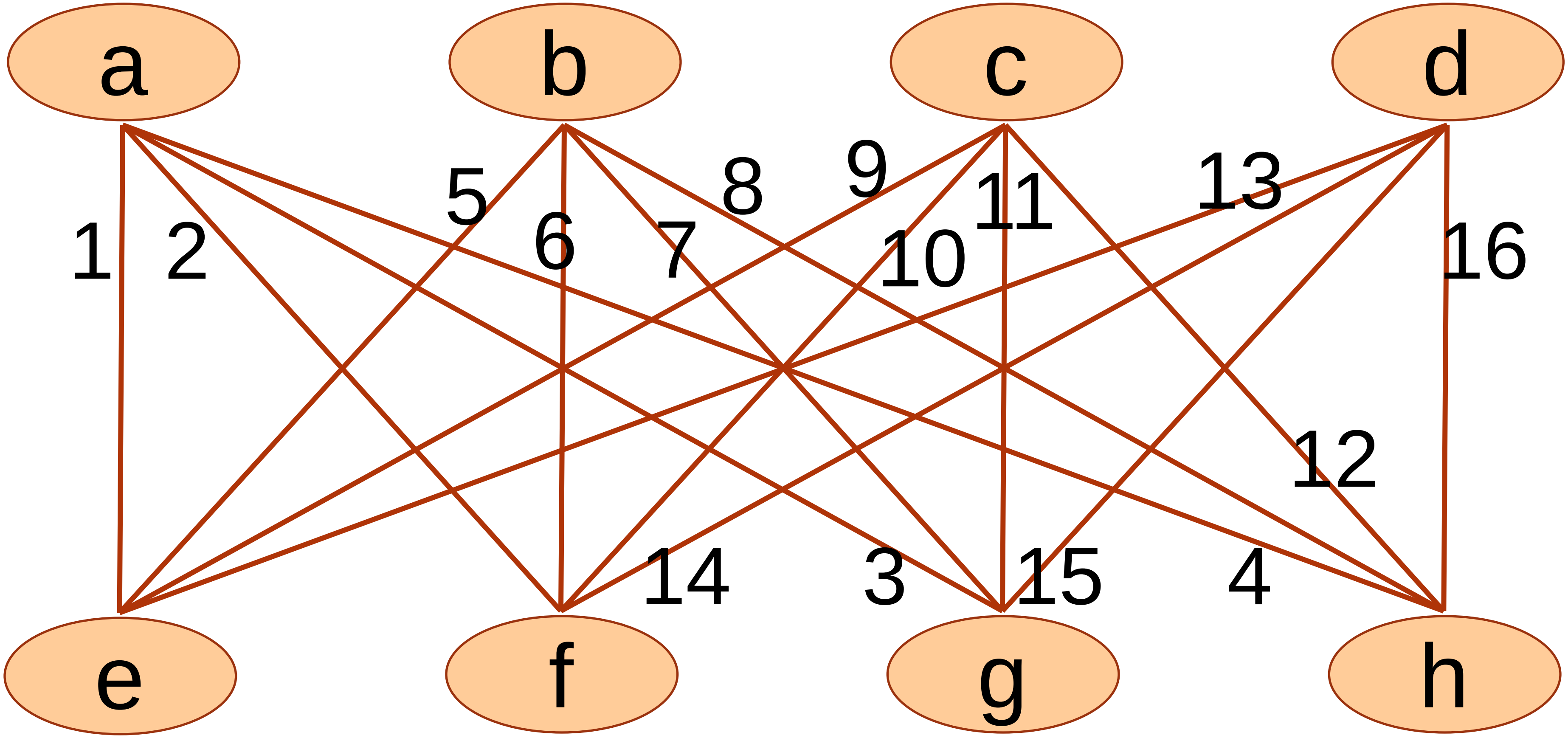
Graph Representation

- Data disklets are the edges of the graph
- Parity disklets are the vertices of the graph
- Reliability stripe is composed of a vertex (parity disklet) and all edges adjacent to vertex (data disklets)

Two-dimensional RAID

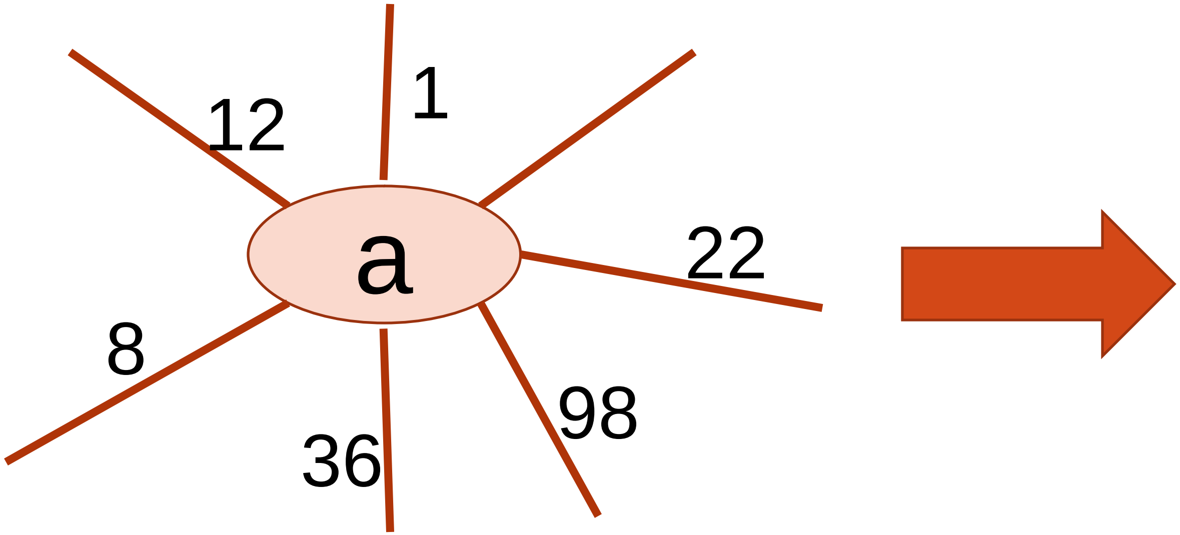


Graph Representation



Graph Representation

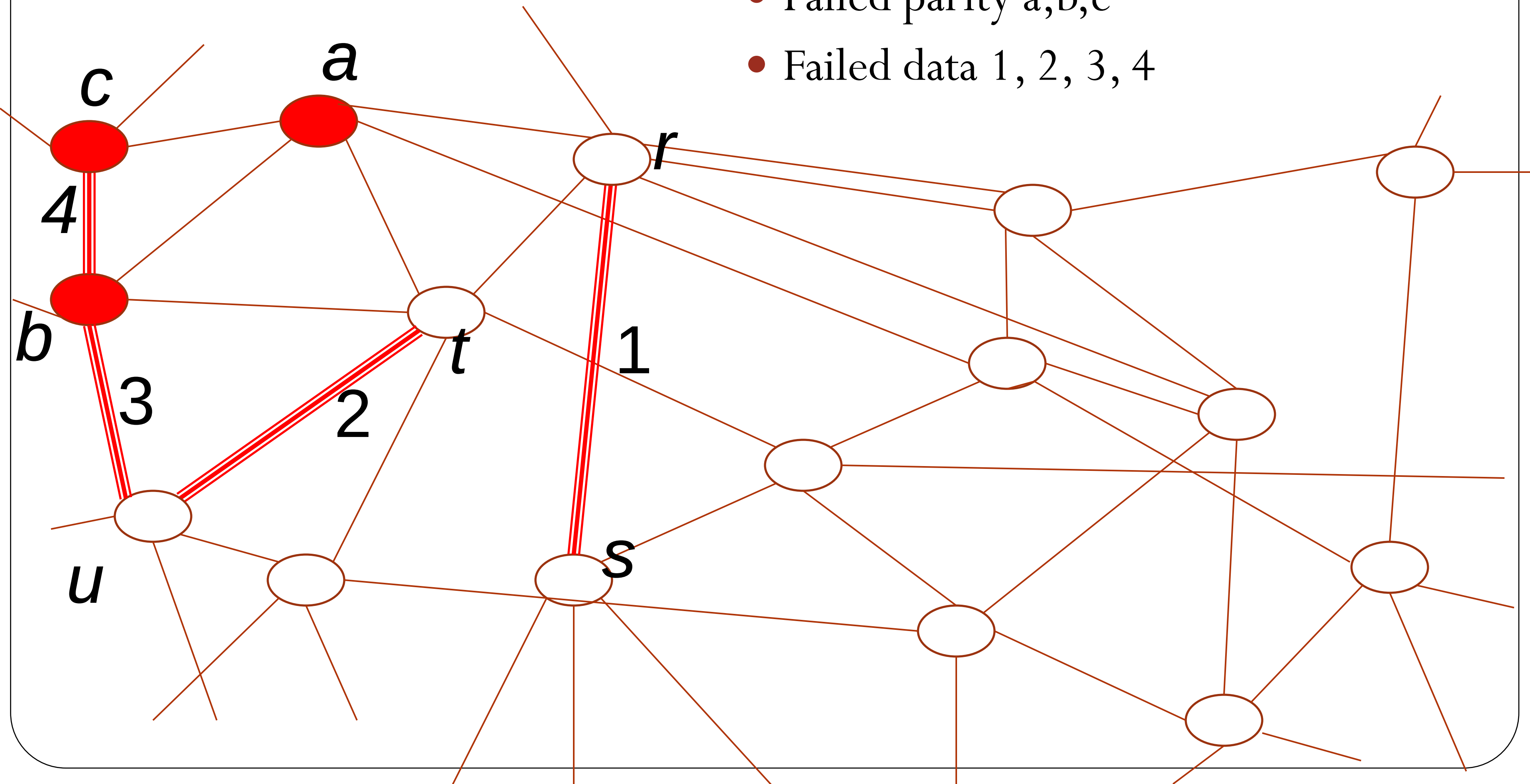
- **Any** graph corresponds to a disklet layout
  - Vertices correspond to parity disklets and reliability stripes
  - Edges correspond to data disklets
  - Adjacency corresponds to reliability stripe membership



Reliability Stripe:
1,8, 12, 22, 36, 98 +
parity a

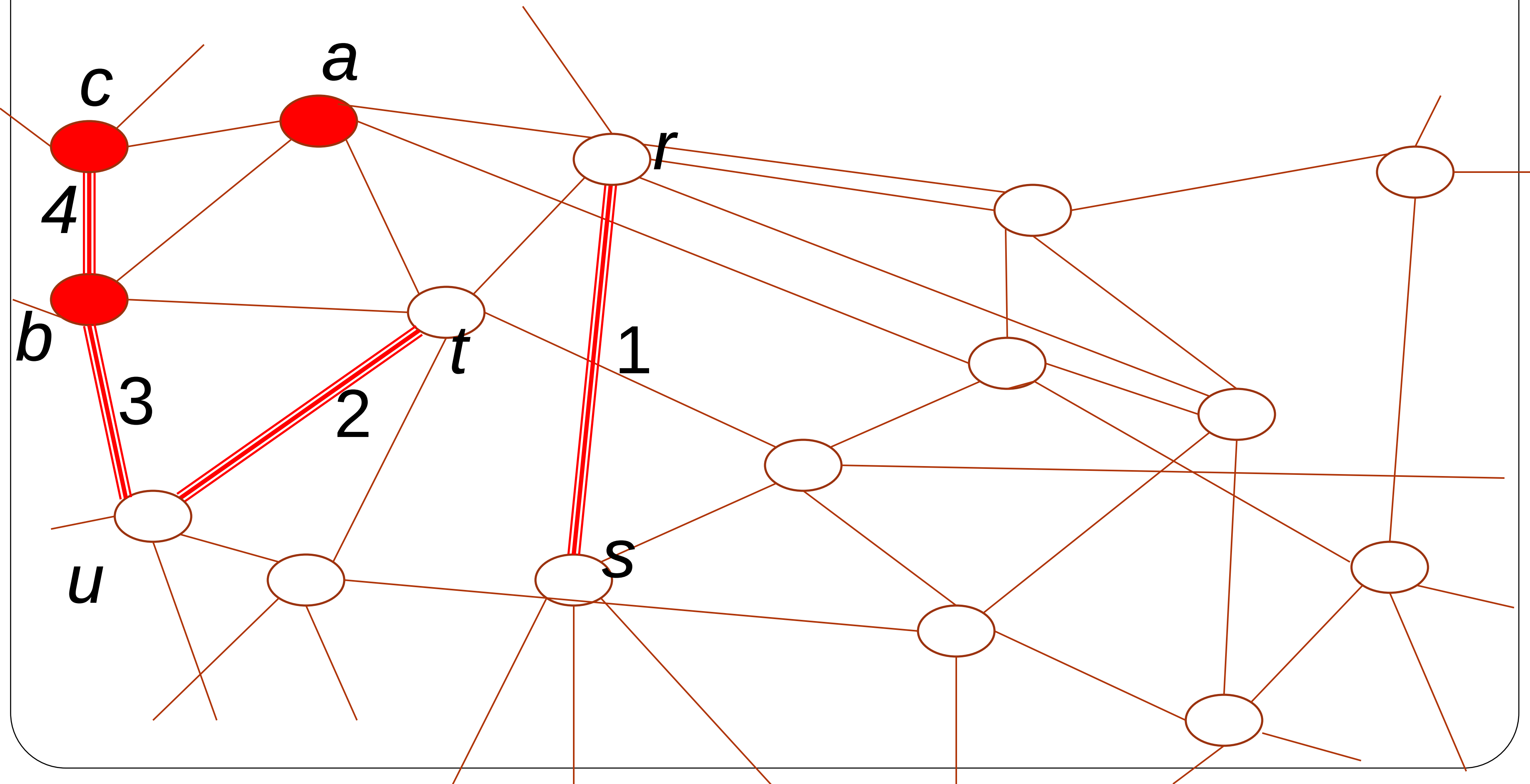
Representing Failures

- Mark failed disklets red:
  - Failed parity a,b,c
  - Failed data 1, 2, 3, 4



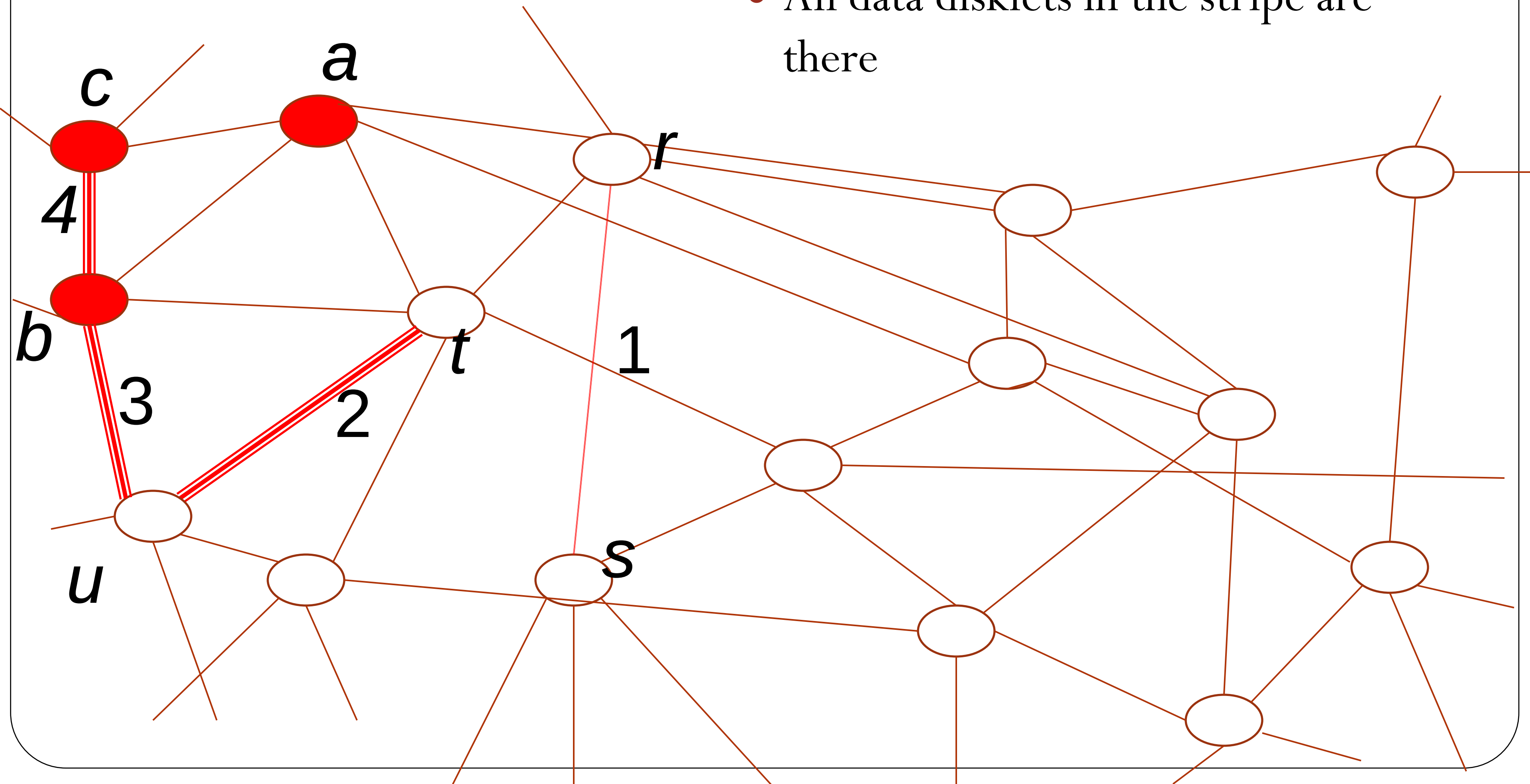
Representing Failures

- Data 1 can be recovered using parity disklet r or s
  - Place on new disklet



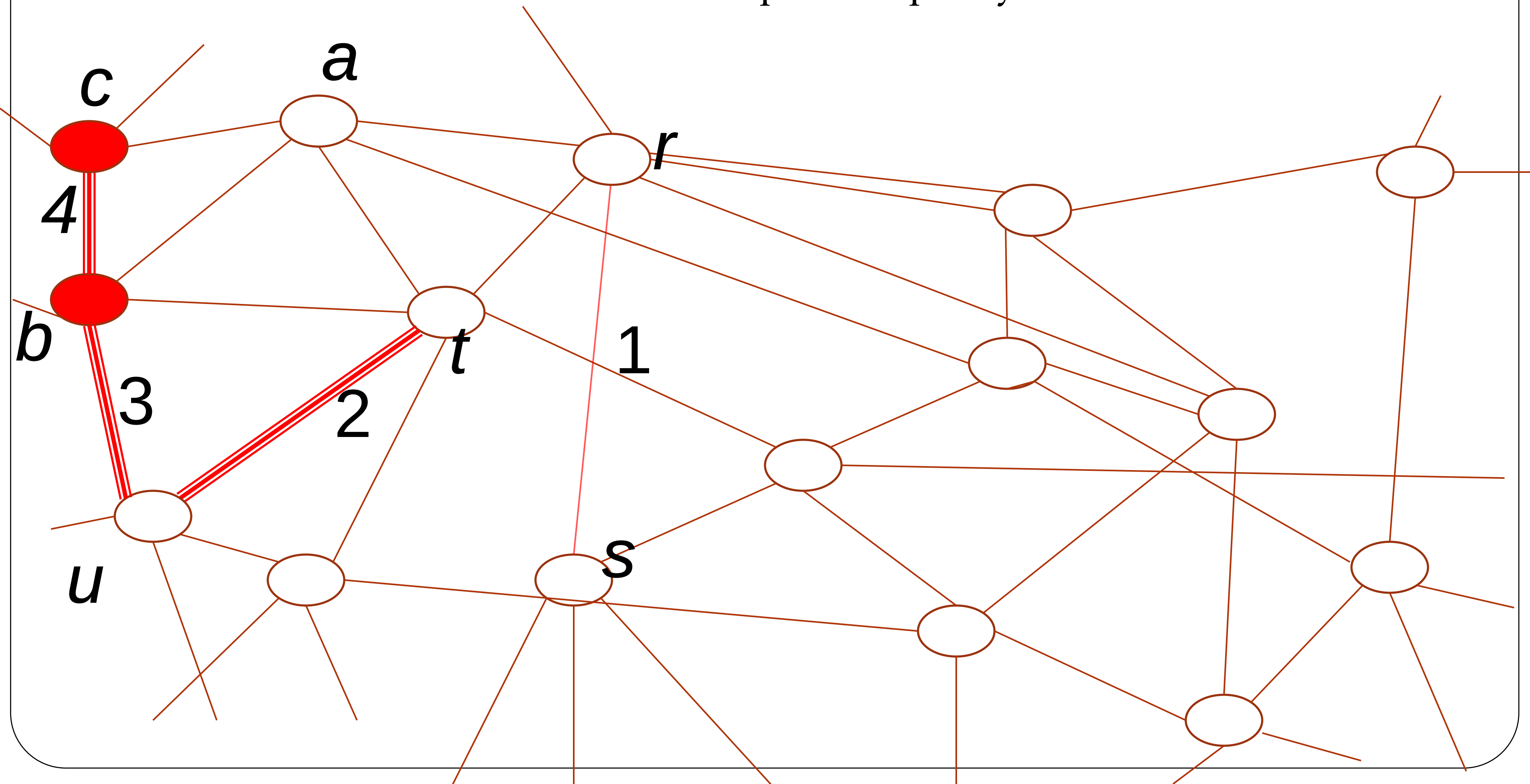
Representing Failures

- Parity disklet a can be recovered
  - All data disklets in the stripe are there



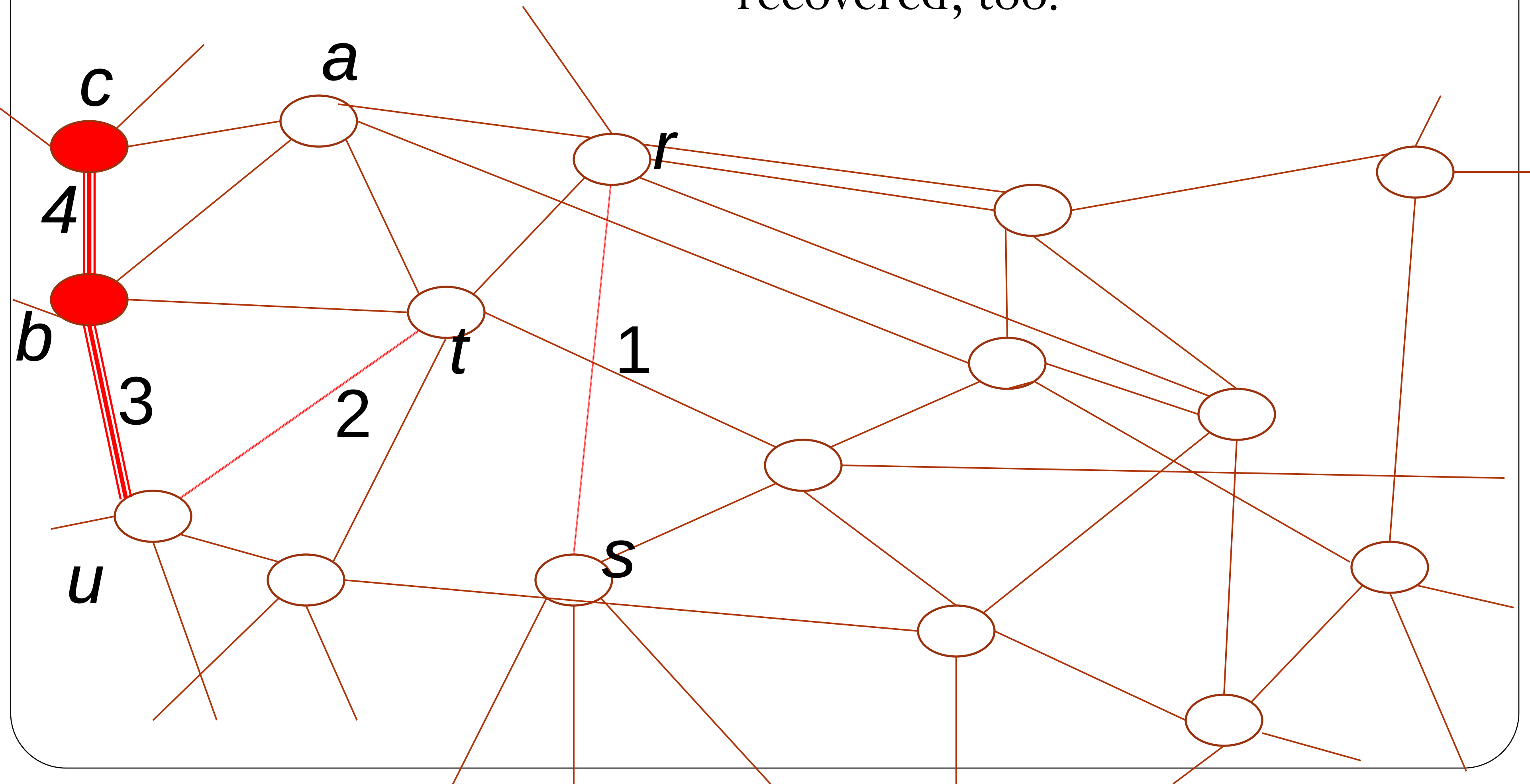
Representing Failures

- Data disklet 2 can be recovered using stripe with parity *t*



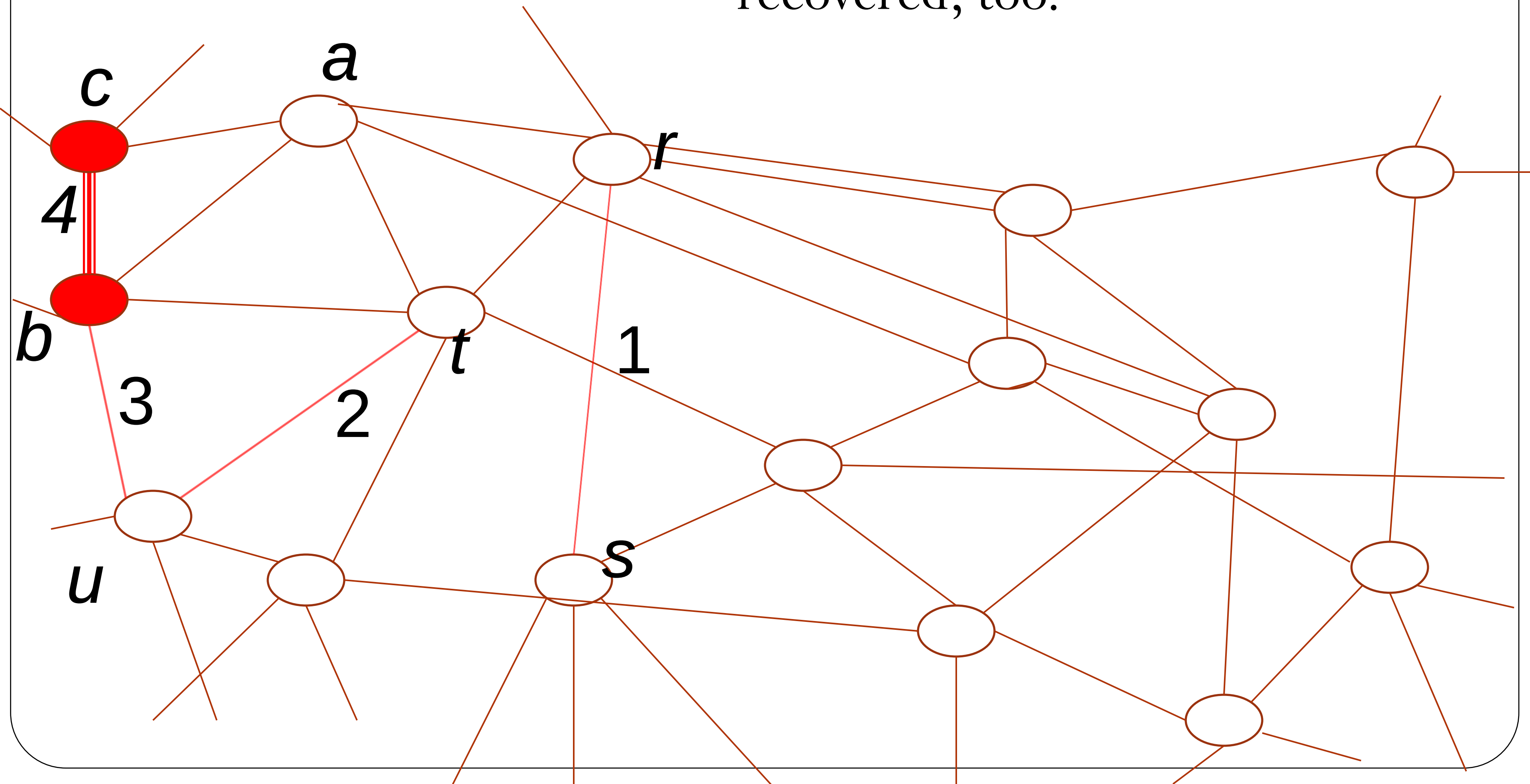
Representing Failures

- Data disklet 3 can *now* be recovered, too.



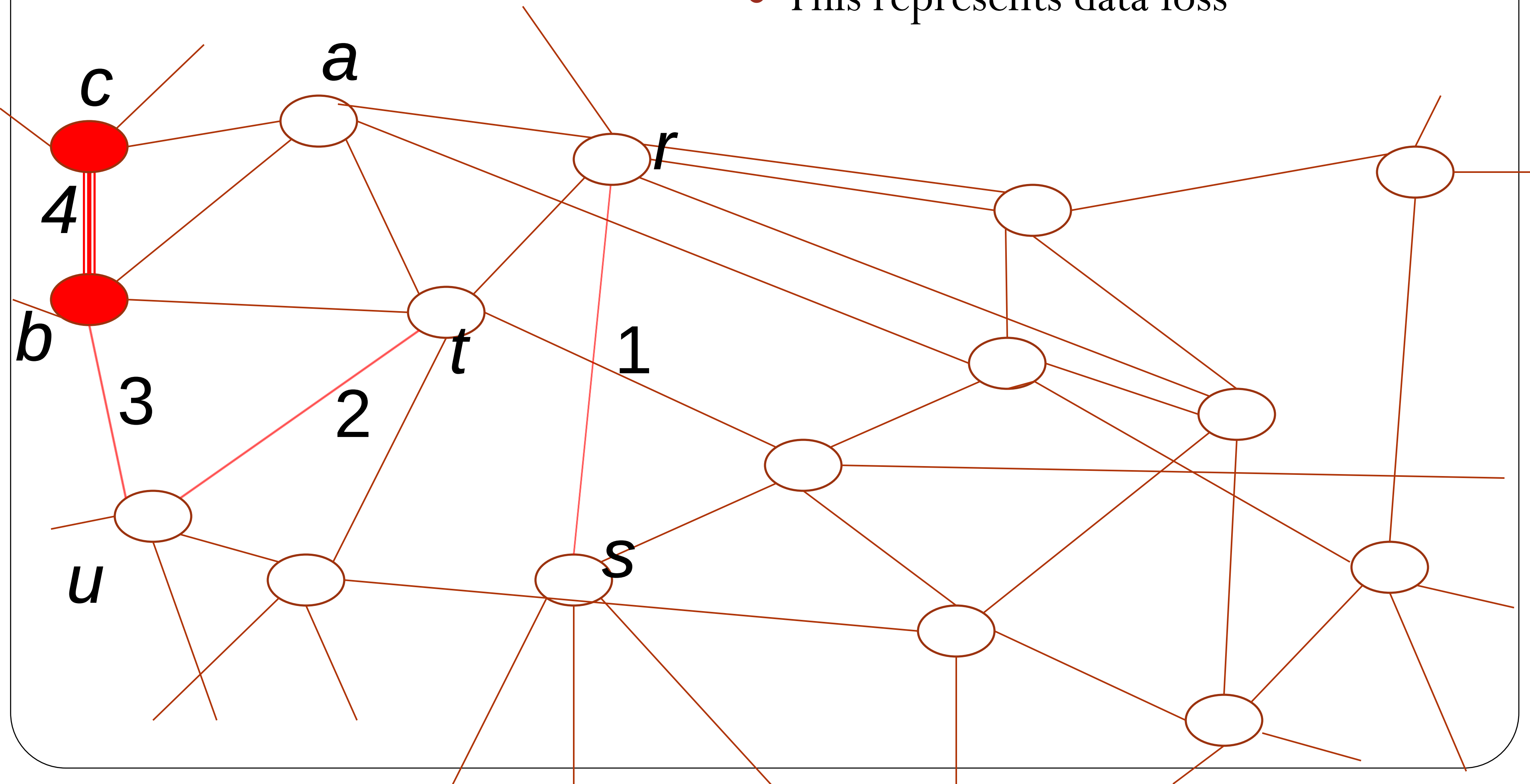
Representing Failures

- Data disklet 3 can *now* be recovered, too.



Representing Failures

- But now we are stuck:
  - This represents data loss



Representing Failure

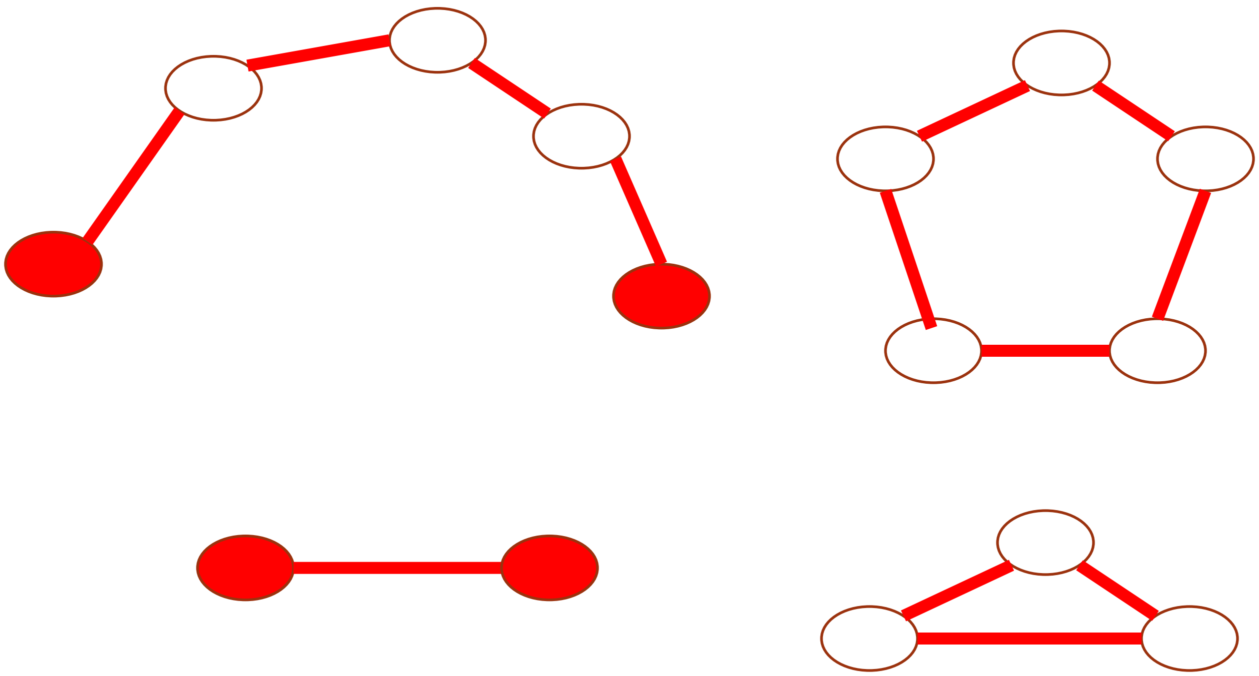
- Disk(s) or rack failure mark(s) many disklets red
    - This is a *failure pattern*
  - Many disklets can be recovered
    - Their data is reconstructed and placed on new disklets
  - Parity disklet (vertex) can be recovered if all edges are not marked failed
  - Data disklet (edge) can be recovered if one of its adjacent vertices and all other edges at this vertex are not marked failed

Representing Failure

- Irreducible failure pattern:
  - Cannot reconstruct (un-mark) any marked edge or vertex
- Minimal irreducible failure pattern
  - An irreducible failure pattern that is not contained in another irreducible failure pattern

Representing Failure

- Theorem: Minimal Irreducible Failure Patterns are:
  - Chains
  - Cycles

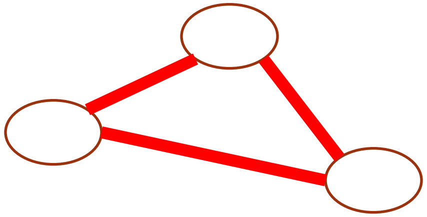


Z. Jie, W. Gang, L. Xiaogugang, and L. Jing, “The study of graph decompositions and placement of parity and data to tolerate two failures in disk arrays: Conditions and existence,” Chinese Journal of Computers, vol. 26, no. 10, pp. 1379–1386, 2003.

Representing Failure

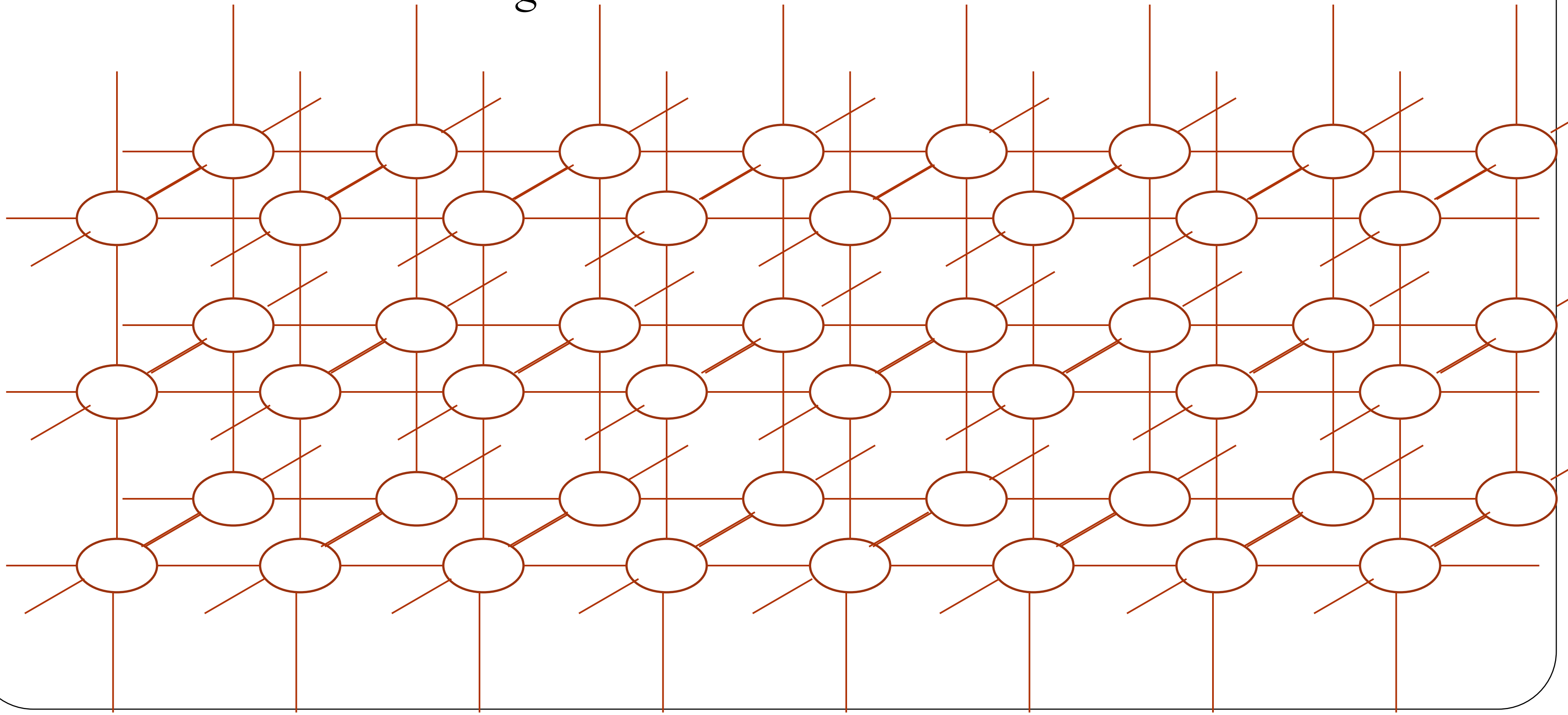
- Not all layouts (graphs) are equal:
  - Cannot avoid the barbell
    - Edges need to be between two vertices

  - But can avoid a triangle



Representing Failure

- We use graphs based on *n*-dimensional grids
  - Guaranteed to be triangle free
  - Have vertex degree = 2 *n*



Assigning disklets to disks

- Disklets need to be stored on disks
  - Simultaneous failure of two disks cannot lead to data loss
- We model this by *coloring* disklets with the color of a disk
  - There are conditions on coloring:
    - To provide two failure tolerance:
      - Every disklet (edge or vertex) needs to be at *walking distance* > 2 of another disklet colored with the same disk
        - Walking distance = Number of elements on the smallest walk connecting two elements
    - This prevents having an irreducible failure pattern generated by a double disk failure

Assigning disklets to disks

- There are conditions on coloring:
  - To provide two failure tolerance:
    - Every disklet (edge or vertex) needs to be at *walking distance* > 2 of another disklet colored with the same disk
  - Every disk should have same proportion of parity and data disklets
  - Reconstruction loads should be evenly distributed
    - In fact, given a massive failure pattern, there are many ways to reconstruct all the data that needs to be reconstructed, as each data disklet is in two reliability stripes
      - *This should follow from our algorithms, but we do not have any results yet*

Assigning disklets to disks

- We use a heuristic / greedy algorithm
  - Line up all disks in a list, then shuffle the list
    - *We call a list of disks a palette*
  - Go systematically through the graph, assigning colors from the list first to vertices, then to edges
    - Check whether walking distance is violated by an assignment, if yes, pick other color, if necessary, backtrack
  - Algorithm guarantees 2 failure tolerance, equal amount of parity

Assigning disklets to disks

- Algorithm works well for racks:
  - Assume that the disk array consists of a reasonably large number of racks, which can fail
  - All disks in a rack are colors in a palette
    - To color an element:
      - First pick a palette (rack) subject to walking distance restriction
      - Then a color (disk) in the palette (rack)

Representing other tasks

- Dealing with massive failure
  - Probably do not have enough spare disklets unassigned in the array
  - Need to **change** graph:
    - Number of data disklets per reliability group needs to be increased so that we need less parity disklets that can then be used to store reconstructed data
  - Changes in the graph correspond to simple operations in the disk array
    - (But these operations move large amount of data from one disk to another)

Representing other tasks

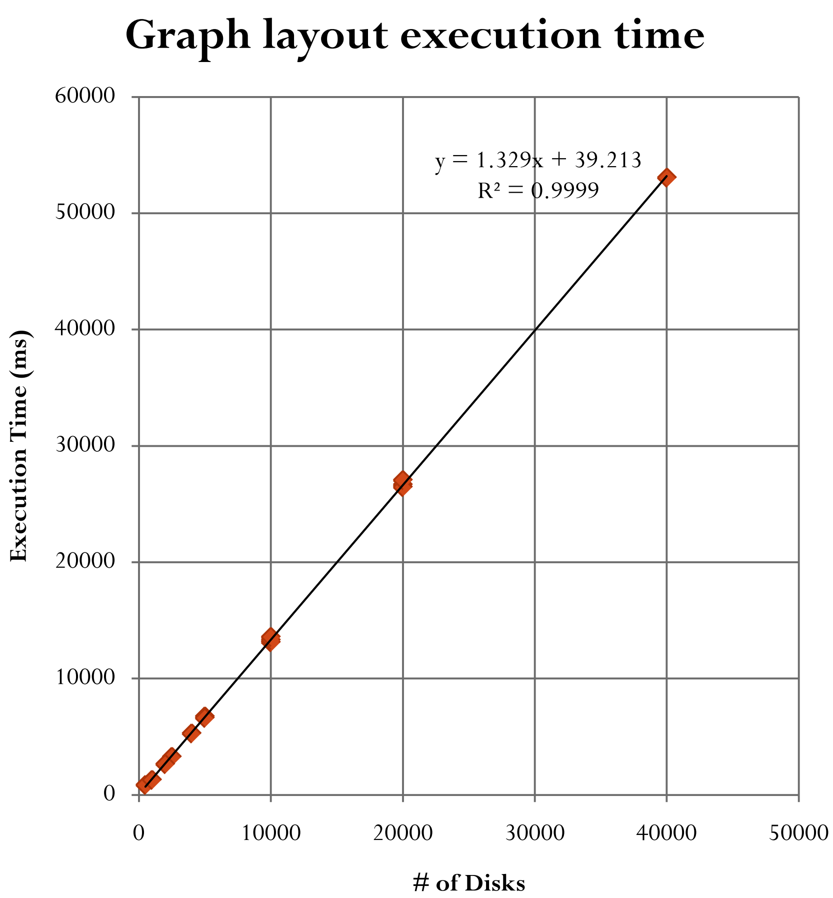
- Moving large amounts of disks into or out of a disk array
  - Corresponds to rather simple graph manipulations

What can we achieve

- Make administration of two-failure resilient, very large disk array ***simple***
- Work in progress:
  - Algorithms need to be fast
  - Need to show that disk layouts are good enough:
    - Resilience against larger sets of failures
    - Distribution of recovery workload

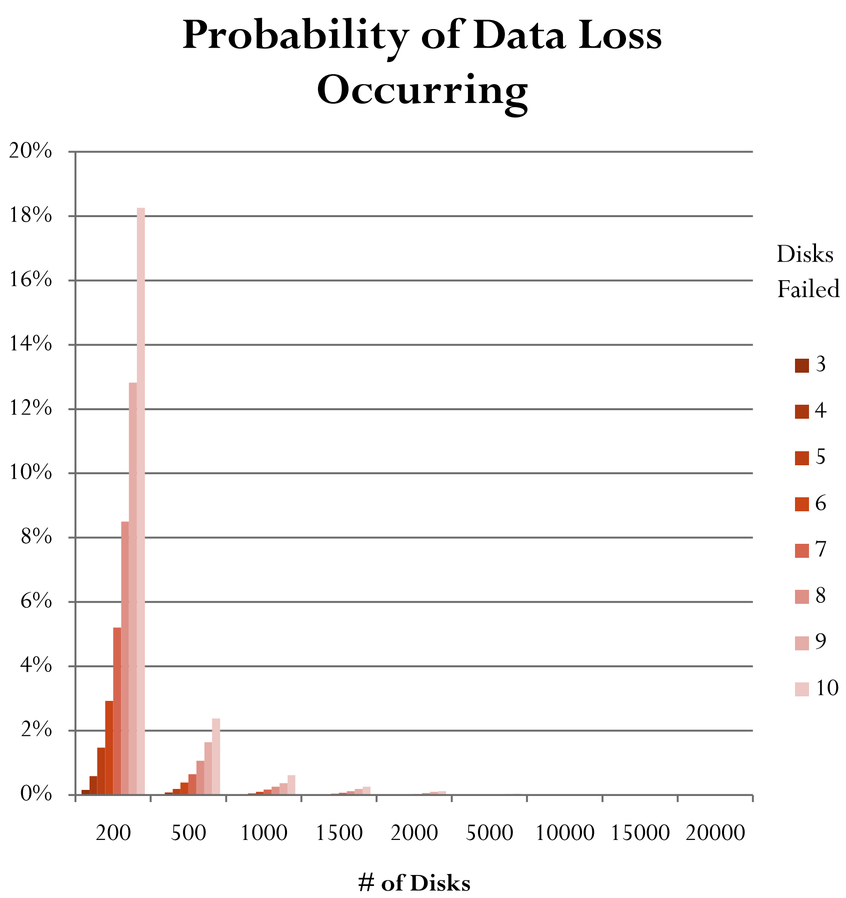
Layout Design : Execution Time

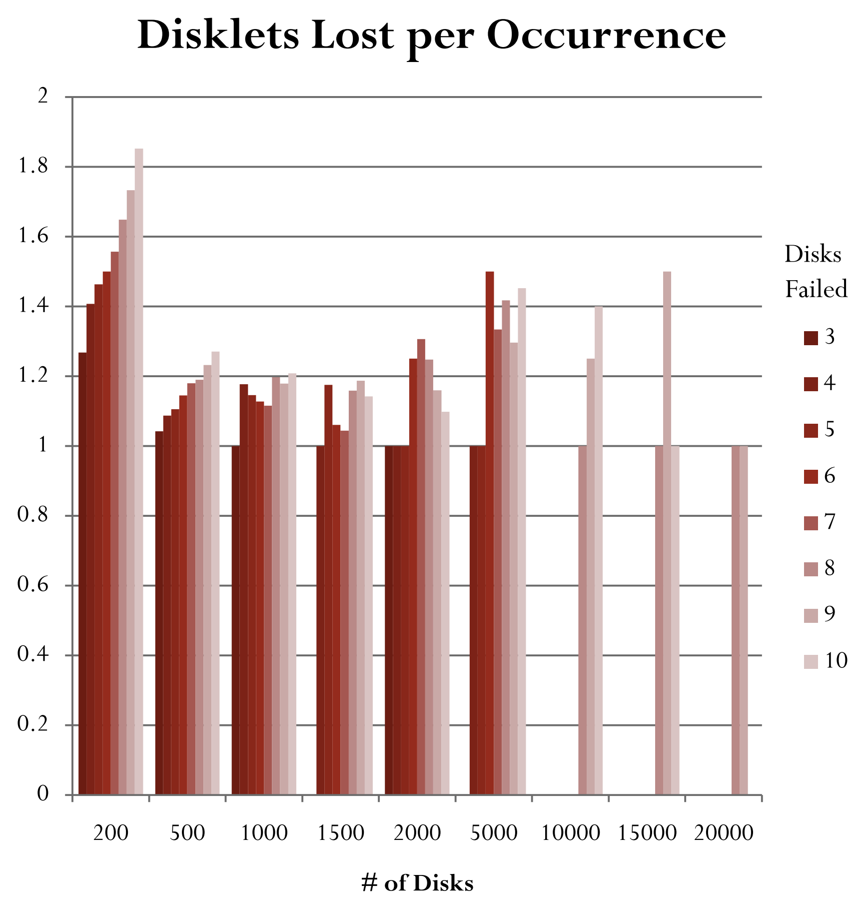
- Graph layout is linear on the number of disks
- Execution time is roughly 1.329ms per disk
- This is **very** fast



System configuration: 10 disklets per Disk, 8 data disklets per reliability group, each data disklet has 2 reliability groups

Layout Design : Failure Tolerance

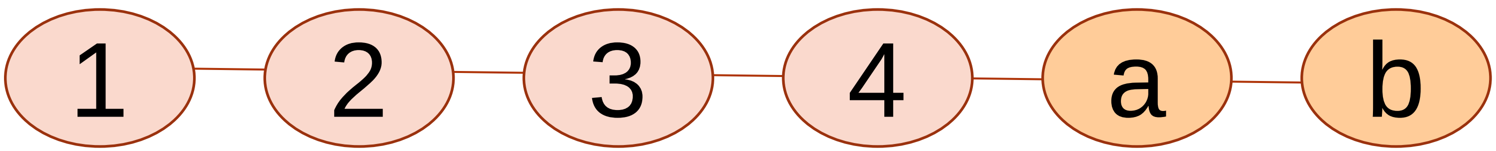




System configuration: 10 disklets per Disk, 8 data disklets per reliability group, each data disklet has 2 reliability groups

Layout Design: Failure Tolerance

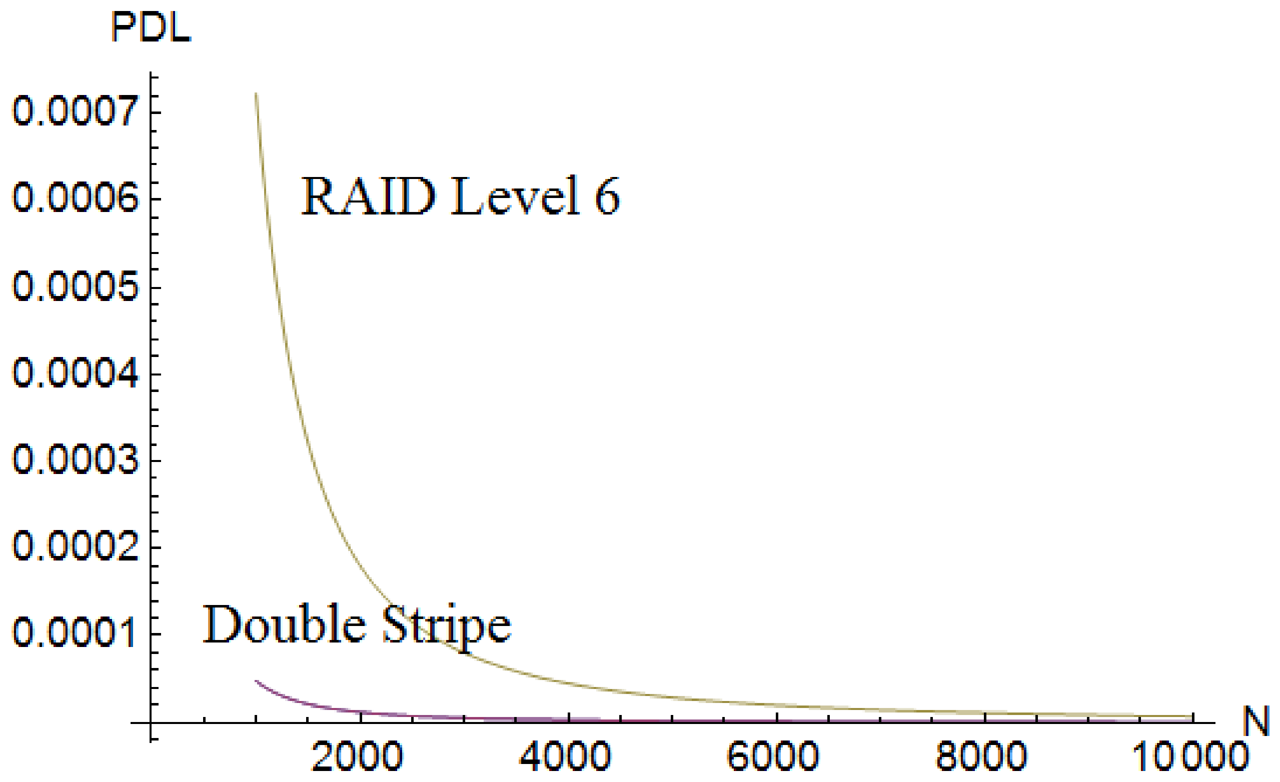
- Alternative: Reliability stripes with two-erasure correcting code (RAID Level 6)



  - Two parity disklets per stripe:
    - One normal parity

  - Has lower robustness

Layout Design: Failure Tolerance



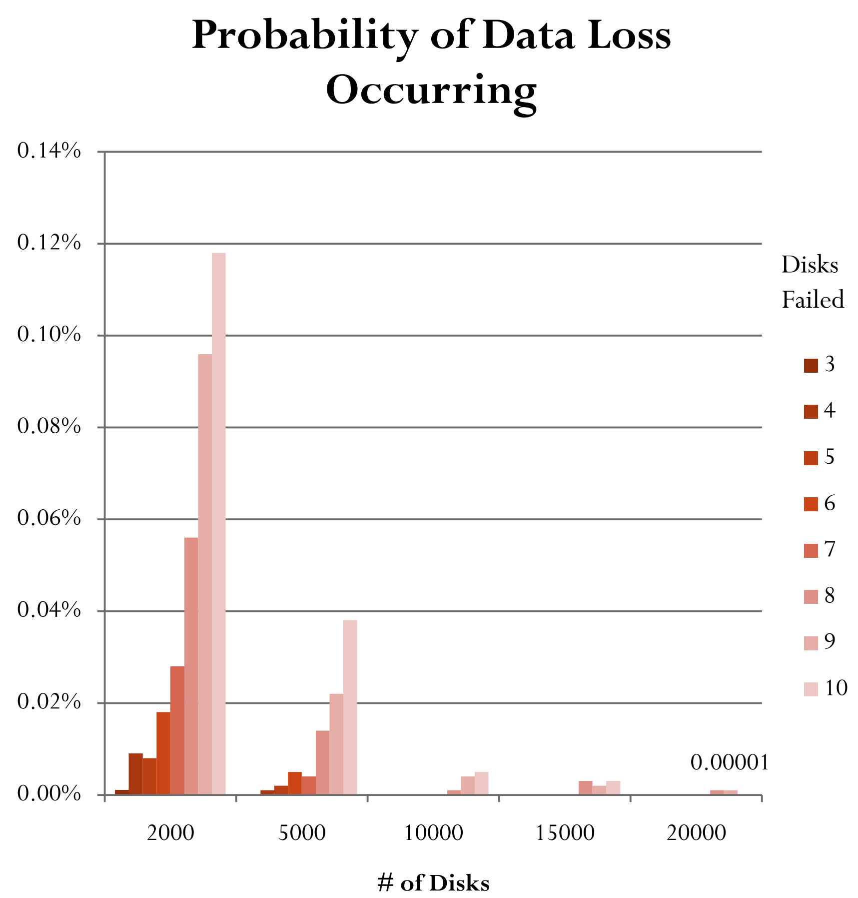
Comparison in Probability of Data Loss (PDL) for three disk failure with 10 disklets per disk between our scheme, below, and RAID level 6 with same storage overhead

Layout Design: Failure Tolerance

- Why is the double stripe strategy more robust:
  - Double stripe with three disk failure:
    - Assume data disklets on one failed disk suffers data loss
      - Then the parity disklets are on the other two disks
  - RAID Level 6 stripe:
    - Assume data disklet on one failed disk suffers data loss
      - If any two of the other disklets in the stripe are on the other two failed disks, we have data loss
      - For $m = 8$, 36 possible failure arrangements.

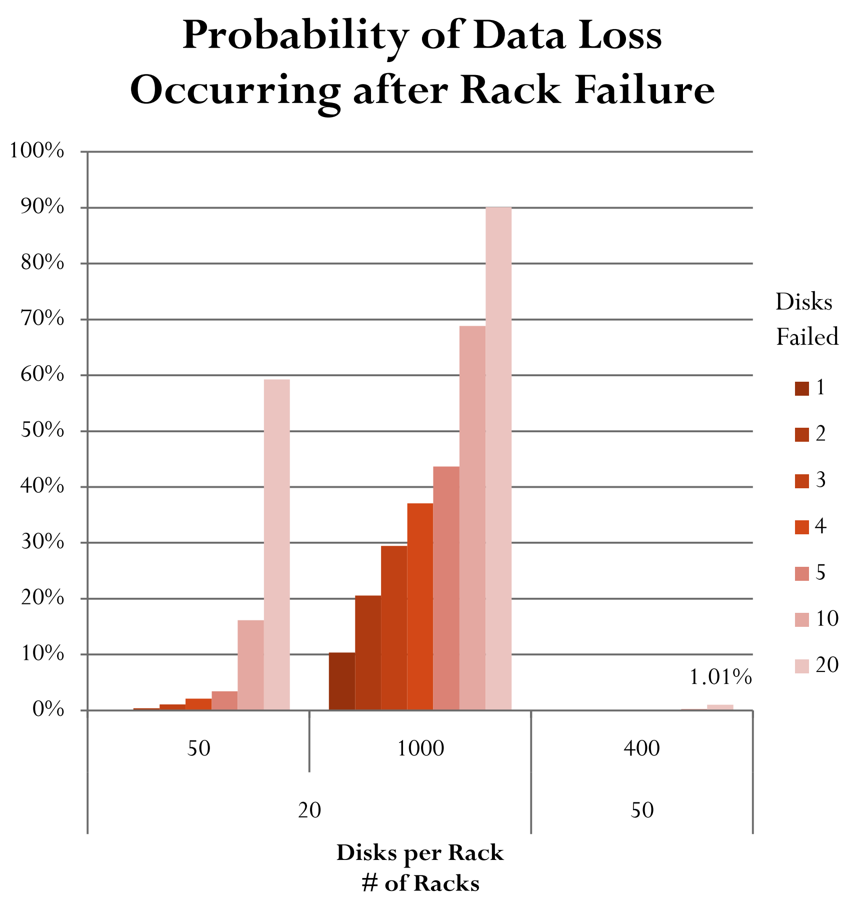
Layout Design : Failure Tolerance

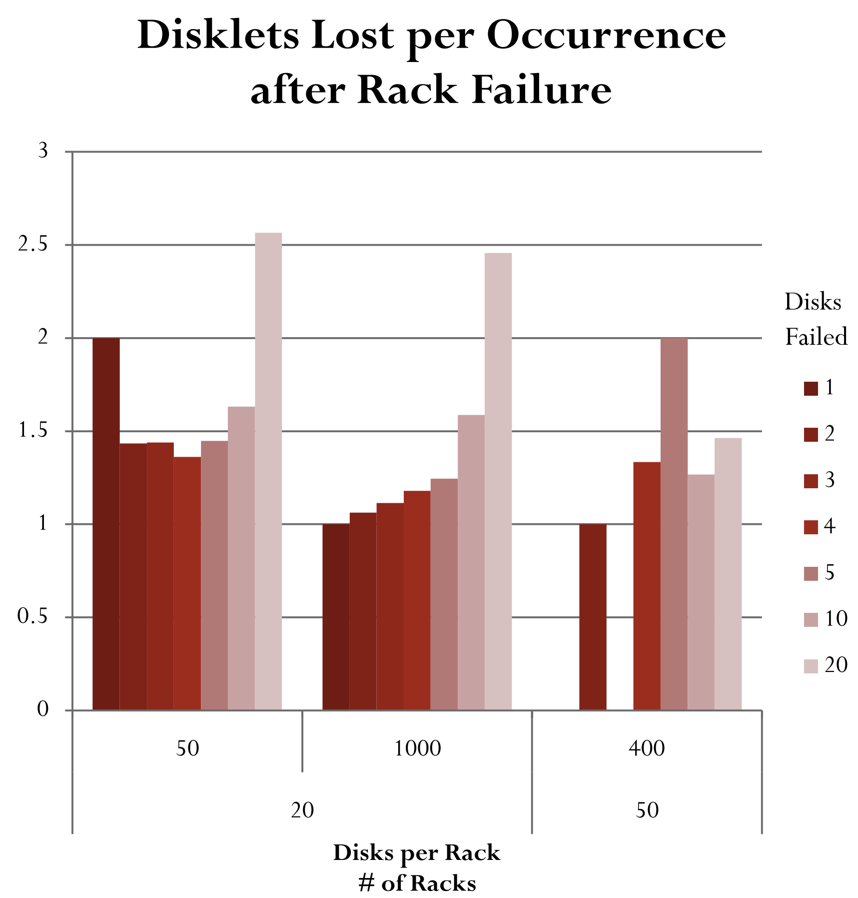
- Failure tolerance increases with the # of disks in the system
- The system can sustain multiple simultaneous disk failures without data loss



System configuration: 10 disklets per Disk, 8 data disklets per reliability group, each data disklet has 2 reliability groups

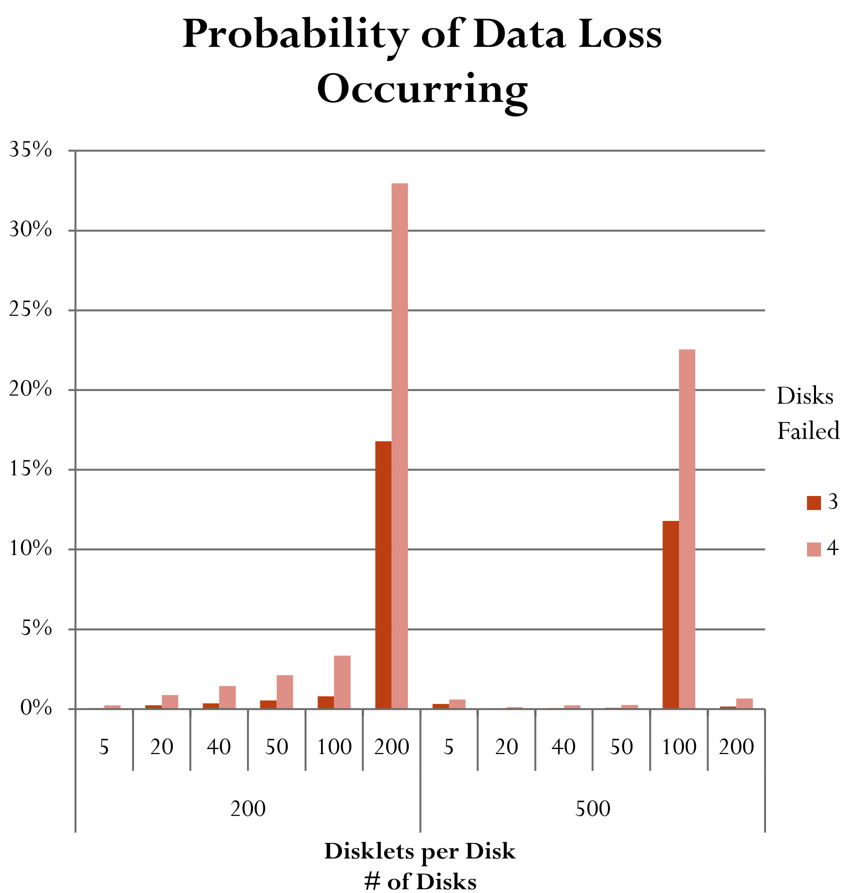
Complete Rack Failure

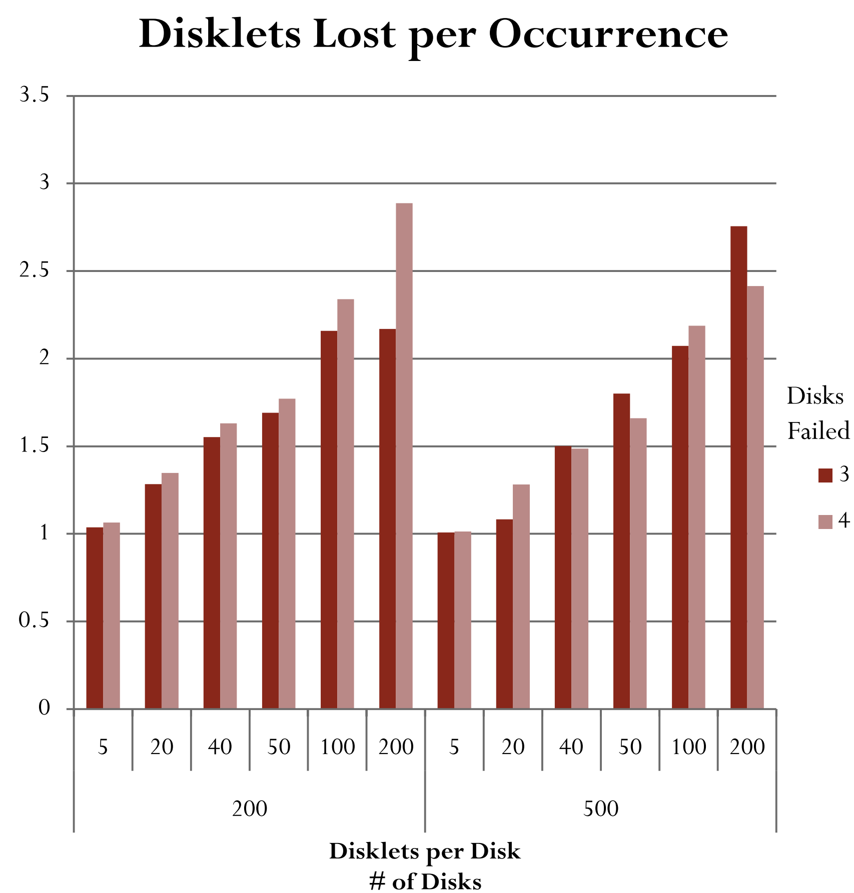




System configuration: 10 disklets per Disk, 8 data disklets per reliability group, each data disklet has 2 reliability groups

of Disklets per Disk : Failure Tolerance

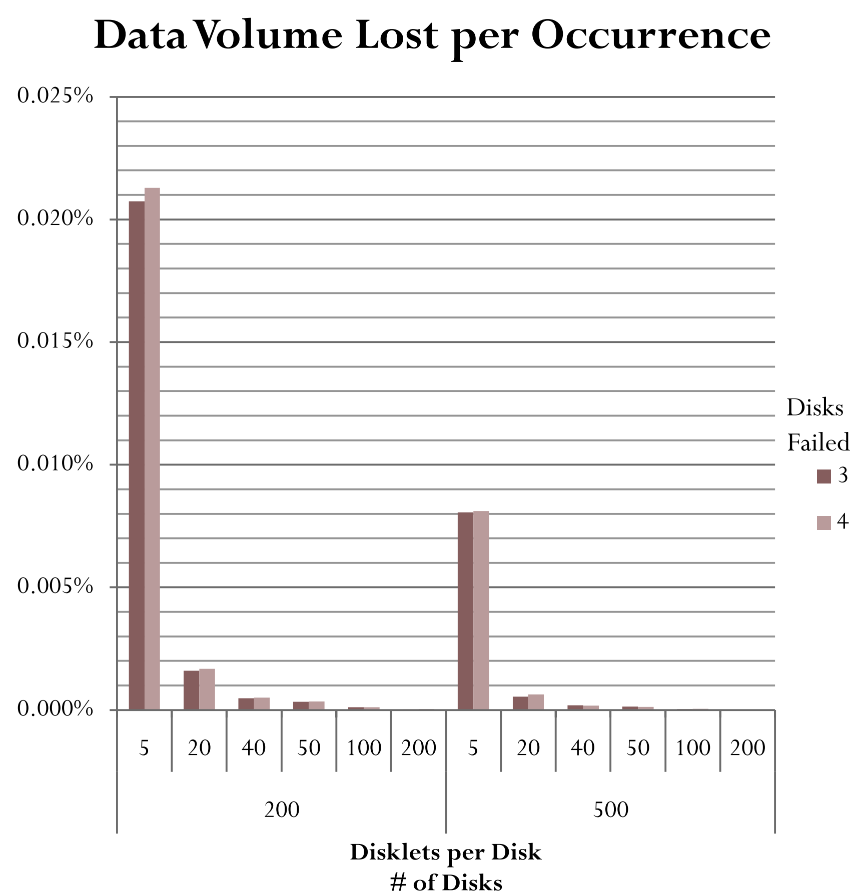




More disklets increase probability of something bad happening at least once
Amount of data lost actually decreases
System configuration: 8 data disklets per reliability group, each data disklet has 2 reliability groups

of Disklets per Disk > Failure Tolerance

- Although the # of units lost increases with the disklets per disk
- The % of actual data lost decreases with the # of disklets per disk



System configuration: 8 data disklets per reliability group, each data disklet has 2 reliability groups