L. Golubchik
Columbia University

Multimedia Storage Servers

Fault Tolerance Issues

Outline

- Background
- Redundancy-based approach (reconstruct)
  - exploit sequentiality of video data delivery
  - disregard sequentiality of video data delivery
- Non-redundancy-based approach (approximate)
  - exploit redundancy that is ‘‘inherently’’ present in video data

Metrics

Resources

- Disk storage
- Disk bandwidth
- Memory buffers

+

Data layout

Scheduling

↓

Quality of service ≡ Latency / Throughput / Reliability

⇨ Don't forget cost!

Need for Fault Tolerance

- Potentially large number of disks, i.e., failure rate is not negligible
  - MTTF of a disk is on the order of 300,000 hrs
    - MTTF of a 1000-disk system is on the order of 300 hrs (or 12.5 days)
- High penalty for unreliable service
  - potentially, long delay in rebuild
- Length and extent of service disruption
  - few hiccups (lasting a few seconds)
  - from minutes to hours of interrupted service

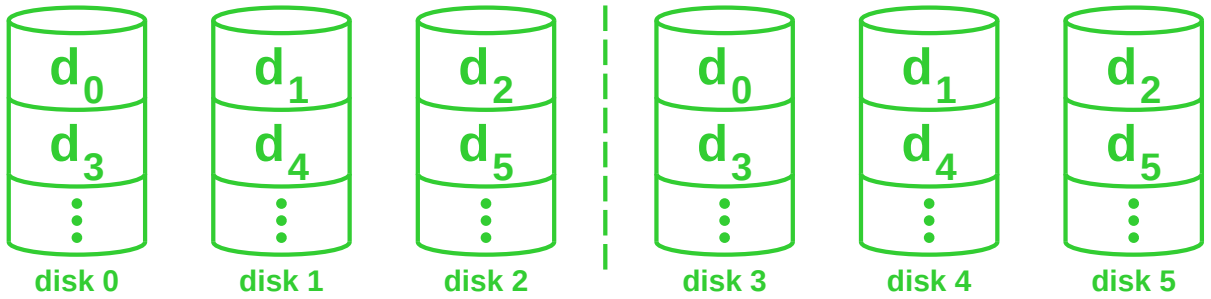
Reliability in Multimedia Servers

- More "stringent" than in traditional database systems
  - real-time recovery from failure
- More "flexible" than in traditional database systems
  - degrees of QoS
    - *"catastrophic failures" (MTTF)*
    - *"degredation of service" (or QoS)*

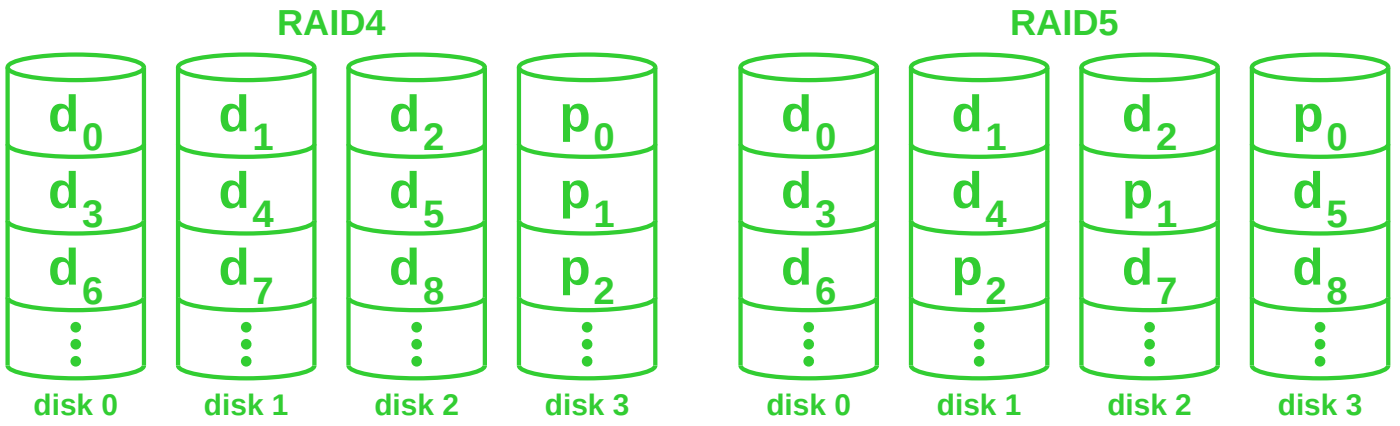
Traditional Approaches

(for disk-based storage)

Full replication



Parity based schemes



Why Not Just Use RAID?

- Traditional RAID solutions are inadequate (for ‘‘real-time’’ applications) due to poor resource usage

- Will discuss alternative schemes which provide a high degree of reliability at
  - lower bandwidth costs
  - lower memory costs

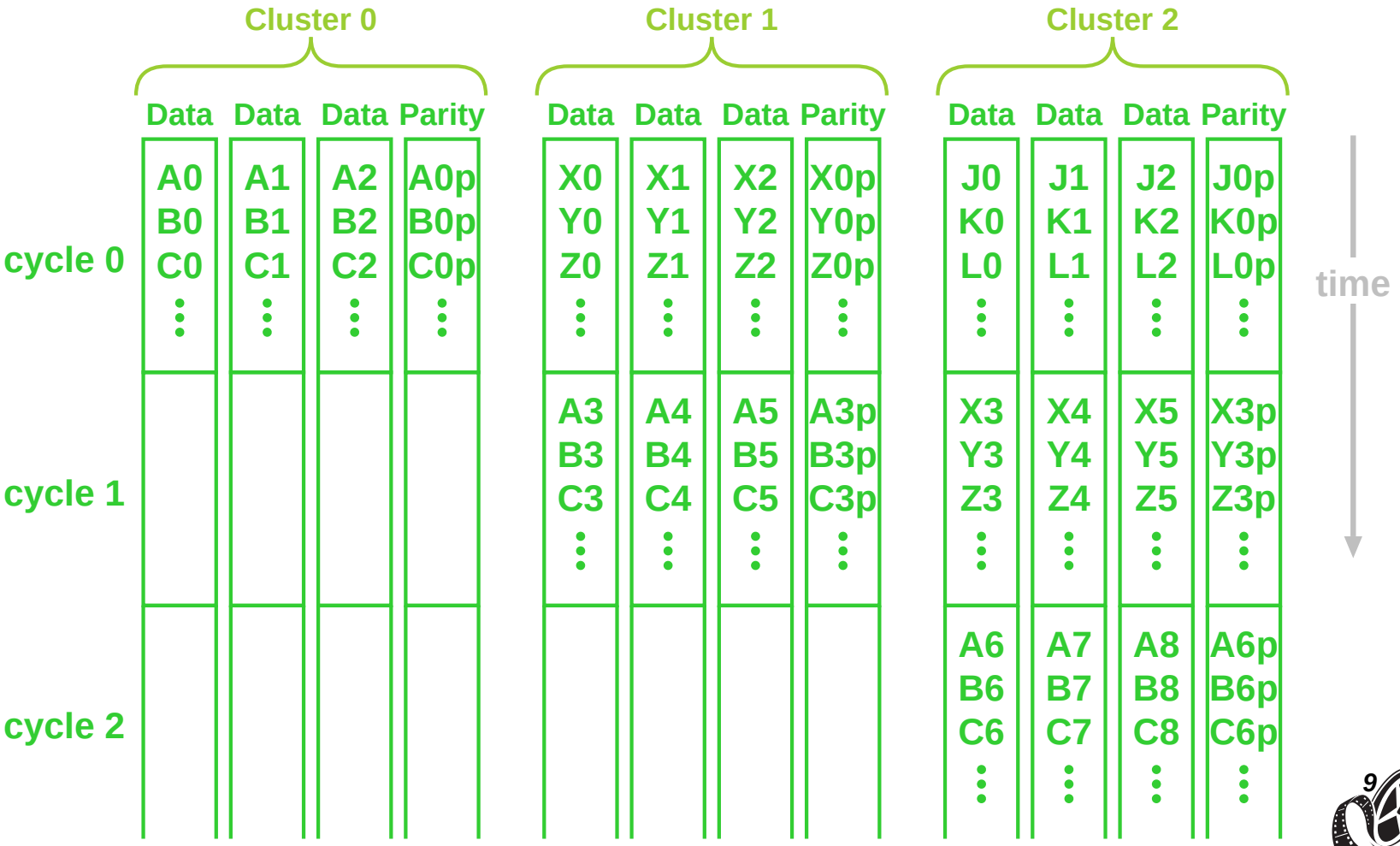
Redundancy-based Sequential Schemes

L. Golubchik
Columbia University

[Tobagi, Pang, Baird, Gang]

Streaming RAID

("traditional" approach)



Advantages of Streaming RAID

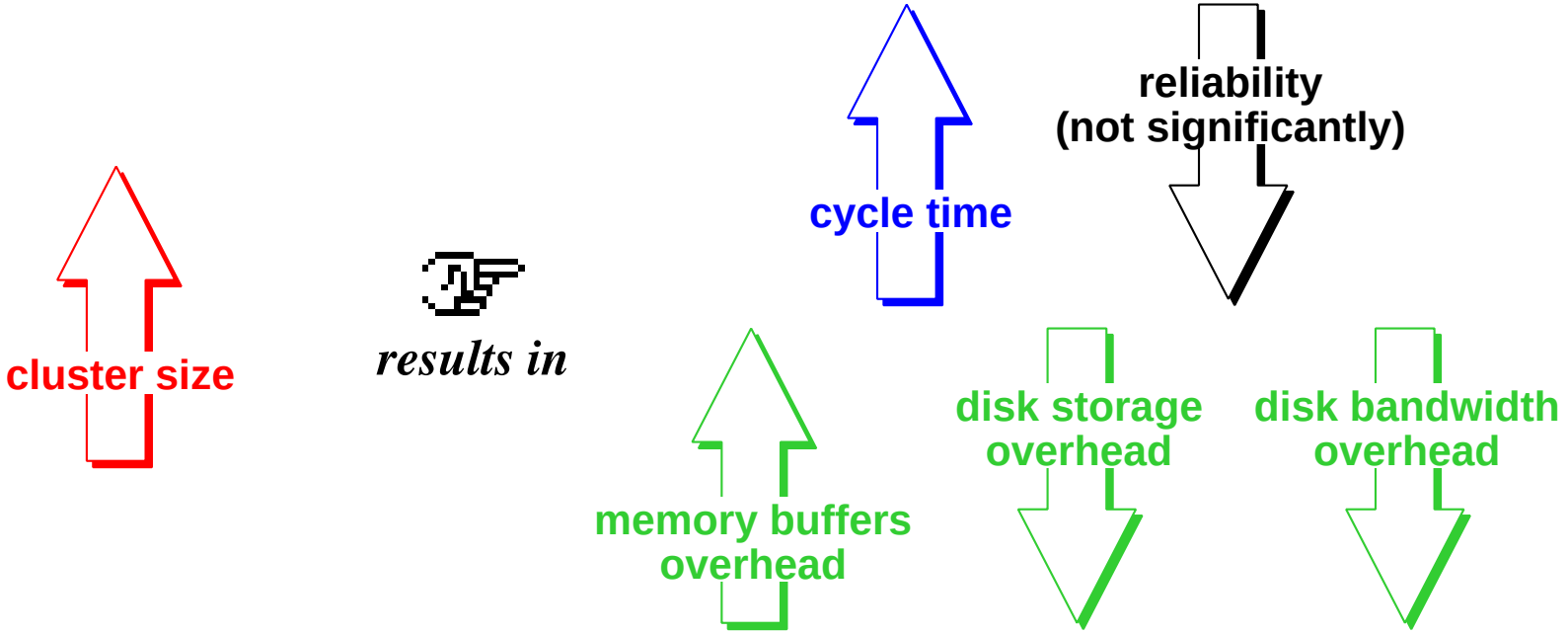
Reliability:

system survives, without hiccups, any set of disk failures such that there is no more than one disk failure per cluster

Disadvantages of Streaming RAID

⇨ **Major disadvantage -- requires a lot of memory**

amount of main memory required for buffering is proportional to the number of streams served per cycle



⇨ **For example, doubling the size of a cluster approx. quadruples the memory requirements of that cluster**

Improving Buffer Space Requirements

(normal operation)

Advantages

- significant decrease in buffer space requirements

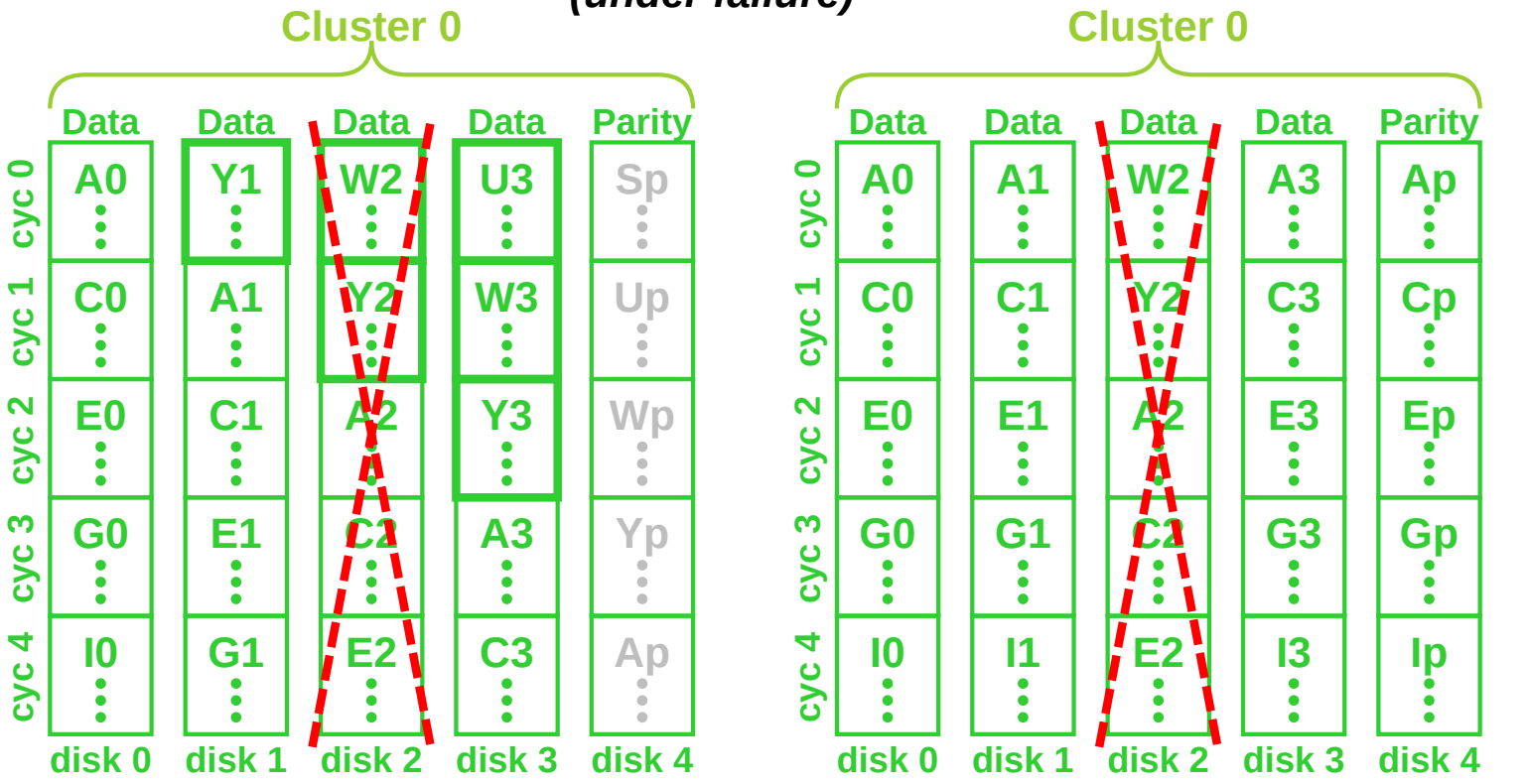
Disadvantages

- when there is a disk failure, there will be hiccups
- *but,* these can be limited, and we can shortly recover to regular service on all streams

[Berson, Golubchik, Muntz]

Impr. Buffer Space Requirements (cont.)

(under failure)



Note: when a cluster is operating under failure, its memory requirement is that of a cluster in the Streaming RAID scheme

Impr. Buffer Space Requirements (cont.)

(under failure)

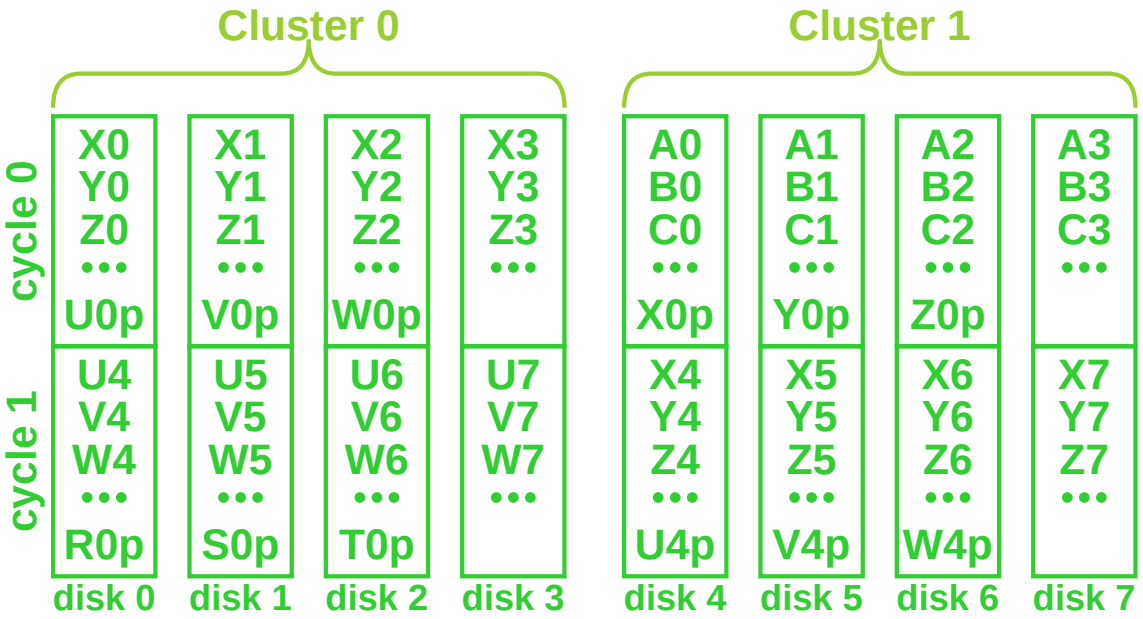
- **Proposal: provide extra memory for a few clusters to be operating under failure**
- **Example:**
  - **100 clusters (of 10 disks each)**
  - **enough memory for 5 to be operating under failure**
  - **mean time till run out of memory is about 250 mil. yrs.**
- **Note that, this memory has to be shared**

[Berson, Golubchik, Muntz]

Improving Bandwidth Requirements

(as applied to Streaming RAID)

a la "Chained Declustering" [Hsiao and DeWitt]



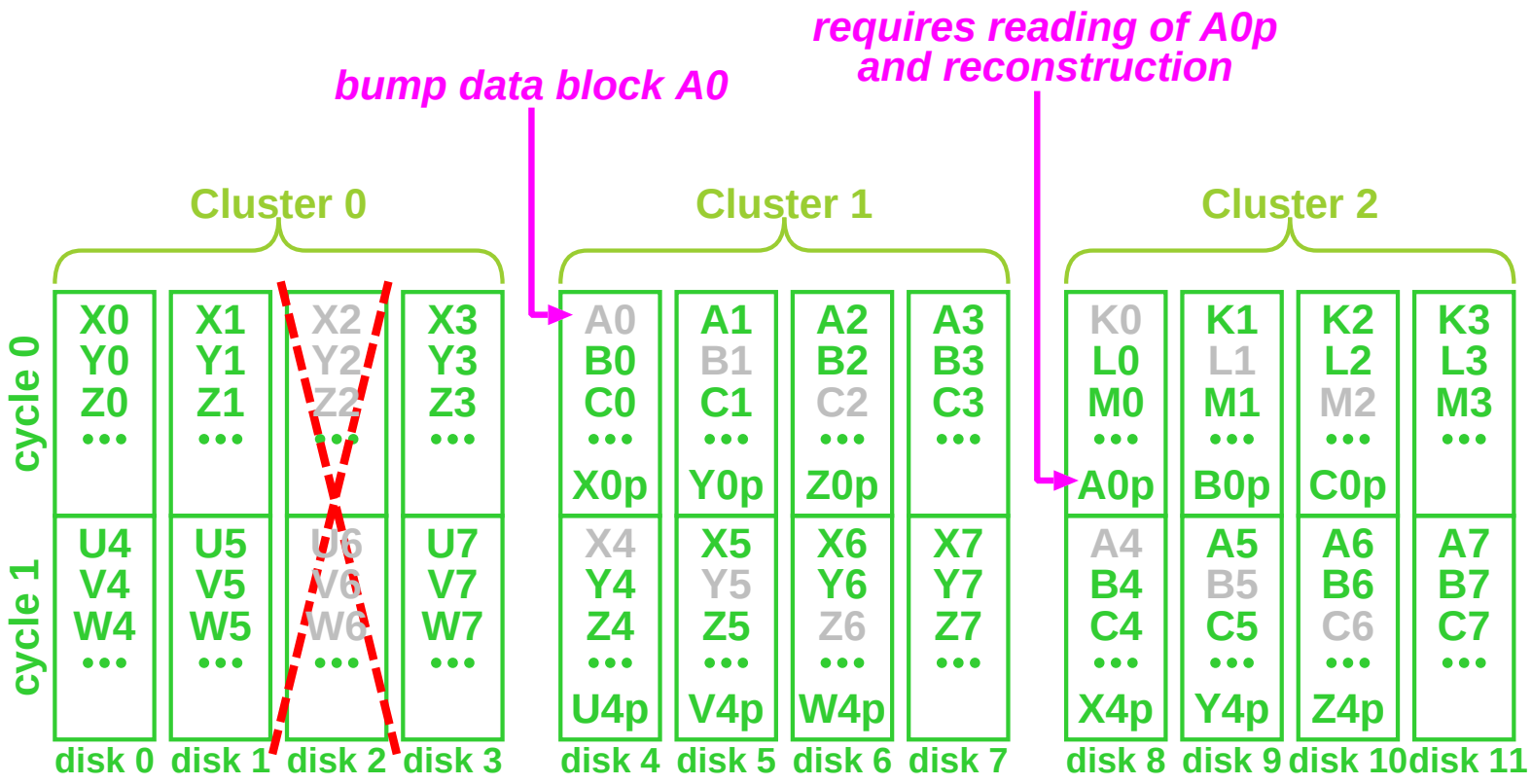
- **Basic idea: rather than dedicate a disk for each cluster to hold parity, distribute the parity for one cluster over the disks of the next cluster** *[degenerate case -- "Chained Declustering"]*

- **Note: during normal operation, we use the bandwidth of *all* disks to deliver data**

[Berson, Golubchik, Muntz]

Impr. Bandwidth Requirements (cont.)

(under failure -- shifting schedule to the right)



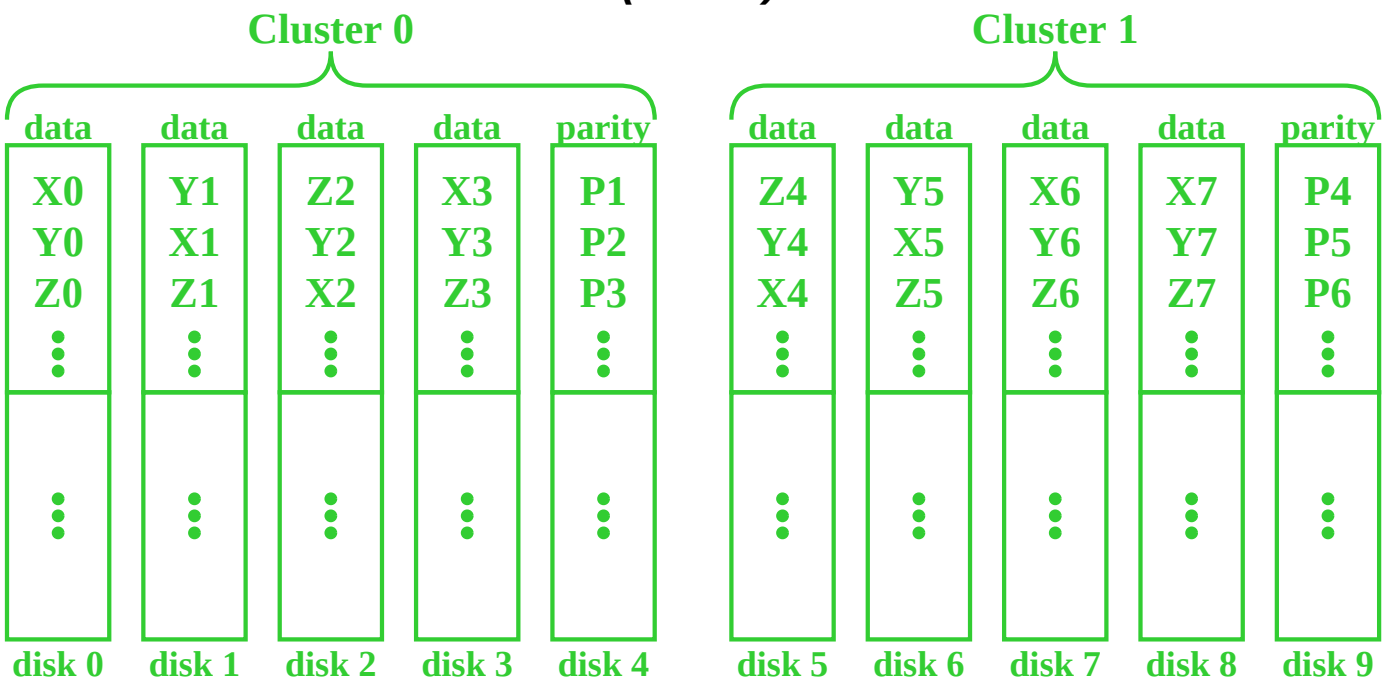
⇨ **Disadvantage: lower reliability**

Redundancy-based Non-sequential Schemes

RAID-based

(RAID5)

[physical layout]



$P1 = X0 \oplus Y1 \oplus Z2 \oplus X3$

$P2 = Y0 \oplus X1 \oplus Y2 \oplus Y3$

$P3 = Z0 \oplus Z1 \oplus X2 \oplus Z3$

$P4 = Z4 \oplus Y5 \oplus X6 \oplus X7$

$P5 = Y4 \oplus X5 \oplus Y6 \oplus Y7$

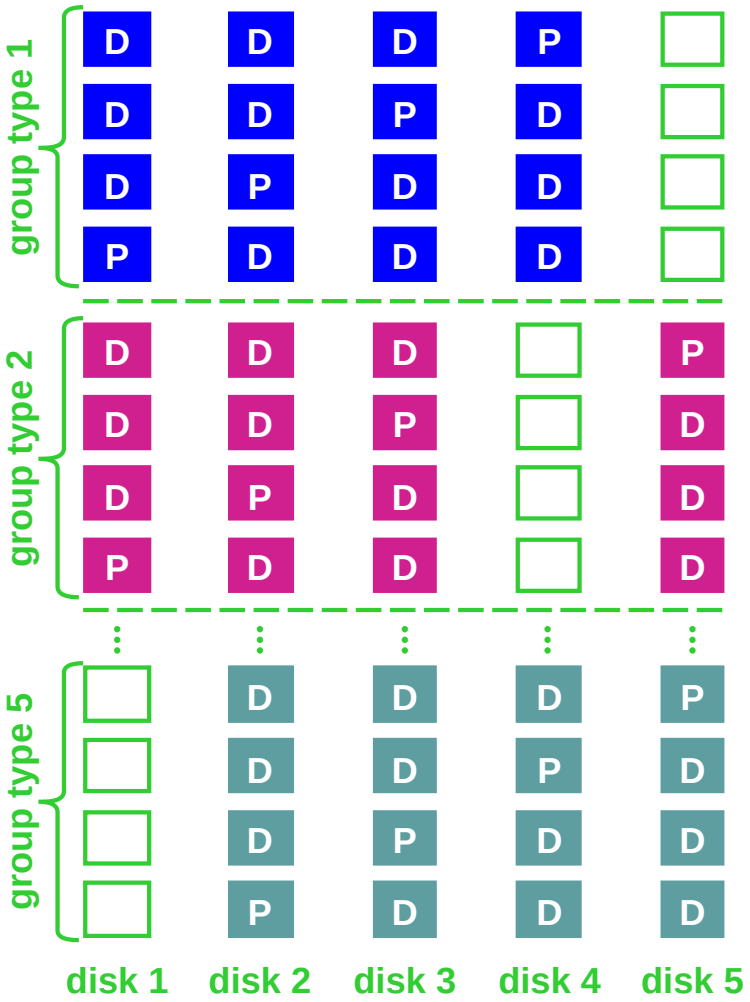
$P6 = X4 \oplus Z6 \oplus Z6 \oplus Z7$

⇨ **Disadvantage: high bandwidth requirements**

[Muntz and Lui]

Improving Bandwidth Requirements

("traditional approach" -- Clustered RAID)



G = 4

C = 5

In case of failure, each disk in a cluster needs to read $x\frac{G-1}{C-1}$ bytes of data as compared to x

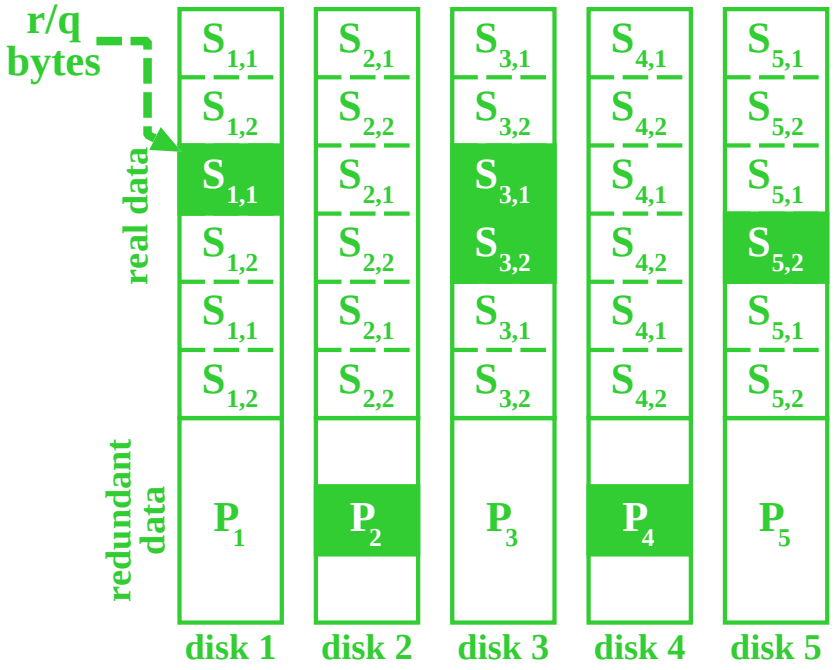
as in the case of RAID5

[Cohen and Burkhard]

Improving Bandwidth Requirements

(SID)

a la "Clustered RAID" [Muntz and Lui]



$$P_1 = S_{5,1} \oplus S_{2,2}$$
$$P_2 = S_{1,1} \oplus S_{32}$$
$$P_3 = S_{2,1} \oplus S_{4,2}$$
$$P_4 = S_{3,1} \oplus S_{5,2}$$
$$P_5 = S_{4,1} \oplus S_{1,2}$$

$q = 2$

$C = 5$

⇨ In case of failure, to reconstruct r bytes of data, each disk in a cluster needs to read at most r/q ***continguous*** bytes from any disk in a cluster, instead of r as in the case of RAID5

[Ozden, Rastogi, Shenoy, Silberschatz]

Improving Bandwidth Requirements

(BIBD)

a la "Clustered RAID" [Muntz and Lui]

- **Specifies a way of constructing s parity groups of size k where every pair of disks occurs in exactly λ different parity groups and each disk occurs in exactly r different parity groups, and total number of disks is u -- then:**

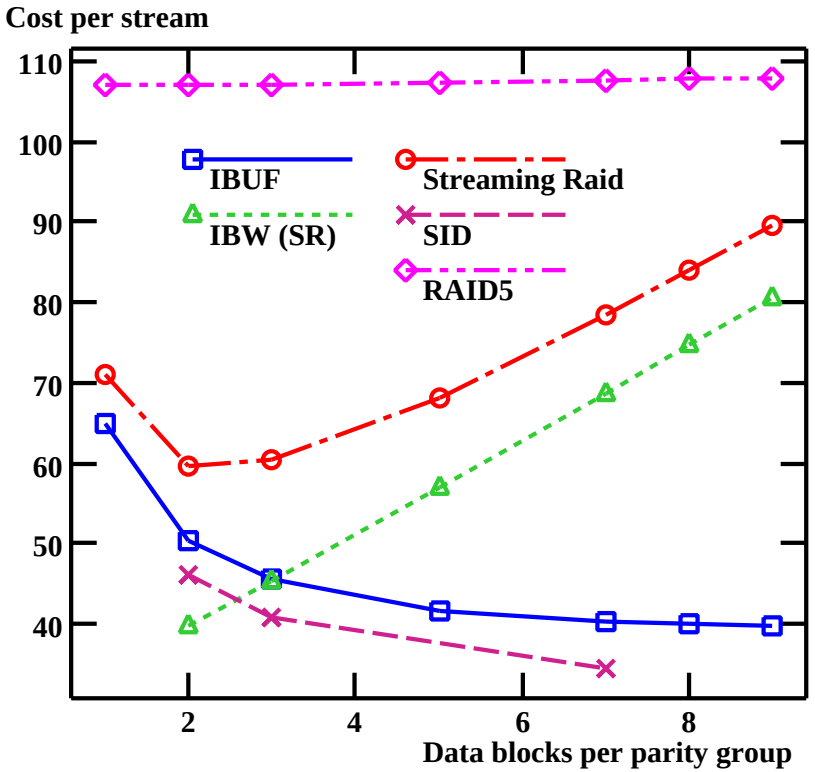
$$r * (k-1) = \lambda * (u-1)$$
$$s * k = u * r$$

- **In case of failure, if N_d is the maximum number of active streams on a disk, then $N_d * r * \frac{k-1}{u-1}$ additional blocks of data will have to be read by each surviving disk instead of $N_d * b$**

as in the case of RAID5

L. Golubchik
Columbia University

Comparison of Schemes

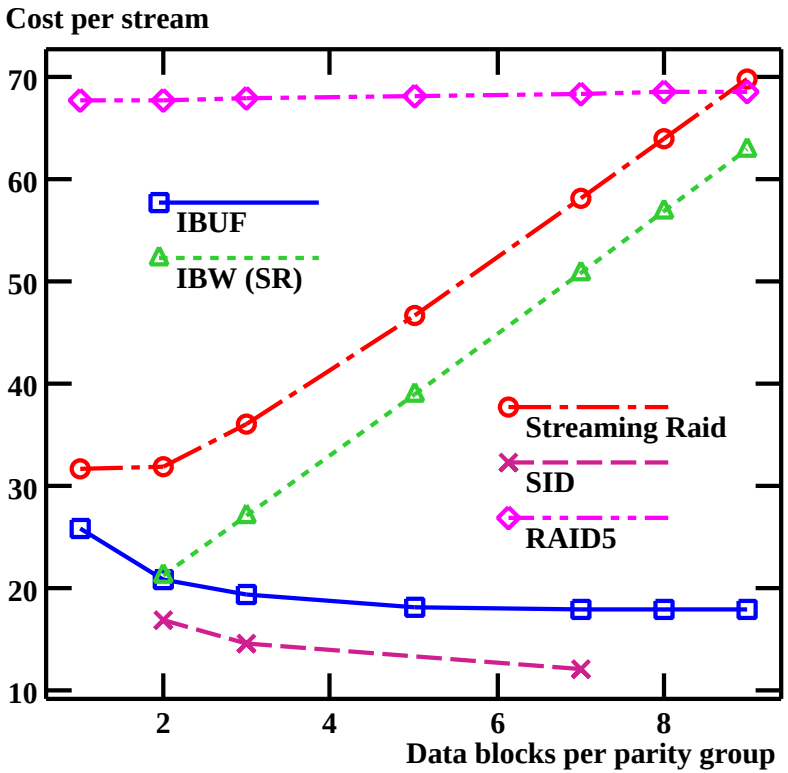


Parameter	Value
b_o	3.5 Mb/sec
B	100 KB
τ_{seek}	20 ms
τ_{trk}	20ms
disk cost	\$0.3 per MB
memory cost	\$30 per MB
MTTF(disk)	300,000 hours
MTTR(disk)	1 hour

L. Golubchik
Columbia University

Comparison of Schemes (cont.)

(changes in technology -- lower disk storage cost)

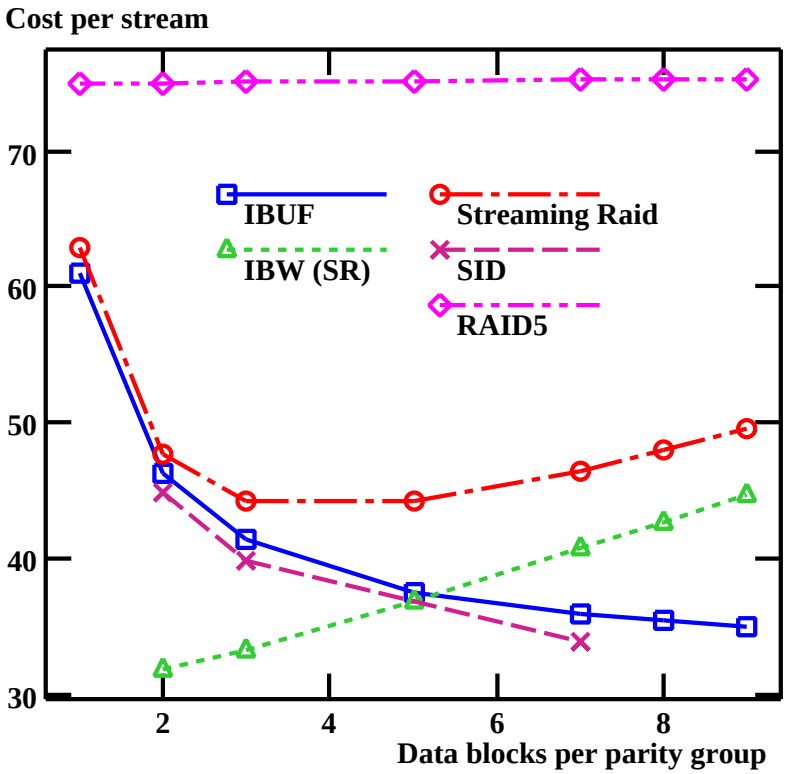


Parameter	Value
b_o	3.5 Mb/sec
B	100 KB
τ_{seek}	20 ms
τ_{trk}	20ms
disk cost	\$0.1 per MB
memory cost	\$30 per MB
MTTF(disk)	300,000 hours
MTTR(disk)	1 hour

L. Golubchik
Columbia University

Comparison of Schemes (cont.)

(changes in technology -- lower memory cost)



Parameter	Value
b_o	3.5 Mb/sec
B	100 KB
τ_{seek}	20 ms
τ_{trk}	20ms
disk cost	\$0.3 per MB
memory cost	\$10 per MB
MTTF(disk)	300,000 hours
MTTR(disk)	1 hour

General Trends

(redundancy-based schemes)

- **Sequential schemes:**
  - **no additional load due to failiure other than for reading parity information (could be reserved)**
  - **potentially large buffer space requirements**
  - **most techniques for reducing the buffer space requirements and the bandwidth requirements for reading parity information**

- **Non-sequential schemes:**
  - **‘‘reasonable’’ buffer space requirements**
  - **potentially large additional load due to failures**
  - **most techniques for reducing the additional load due to a failure**

Non-redundancy-based Schemes

[Vin, Shenoy, Rao]

Non-parity-based Scheme

(self-recovering JPEG (SRJ) algorithm)

[MPEG extension possible]

- **Basic idea**
  - exploit the redundancy inherently present in video objects to approximate information missing due to a disk failure (without the use of error-correcting codes)
  - rebuild ‘‘normally’’, as a low-priority background workload
- **Image quality and storage system efficiency are a fnc of N (number of disks over which an image is partitioned)**

original

N = 4

N = 16

[Vin, Shenoy, Rao]

Non-parity-based Scheme (cont.)

- Advantages (as compared to the traditional RAID scheme)
  - *clients* can perform recovery
  - can withstand multiple failures *[should be careful with degradation of image quality]*

- Should be careful not to make the reconstruction process "slow", since:
  - poorer image quality during reconstruction process
  - there is a higher probability of second failure, where rebuild could no longer be done from surviving disks

Summary

- Fault tolerance in multimedia storage servers is complicated by necessity of recovery in real time
  - fault tolerance issues should be considered when designing data layout & scheduling schemes
  - otherwise, can result in poor resource usage
- Discussed redundancy-based techniques, which resulted in improvements in:
  - memory buffer space requirements
  - I/O bandwidth efficiency
- Discussed non-redundancy-based technique, which resulted in improvements in:
  - processing required for reconstruction of data
  - I/O bandwidth efficiency
- as compared to a traditional RAID approaches