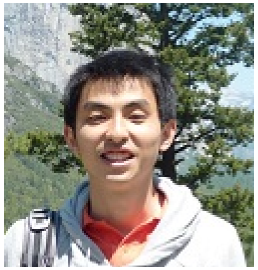
Lazy Means Smart

Reducing Repair Bandwidth Costs in
Erasure-coded Distributed Storage Systems

Mark Silberstein
Technion

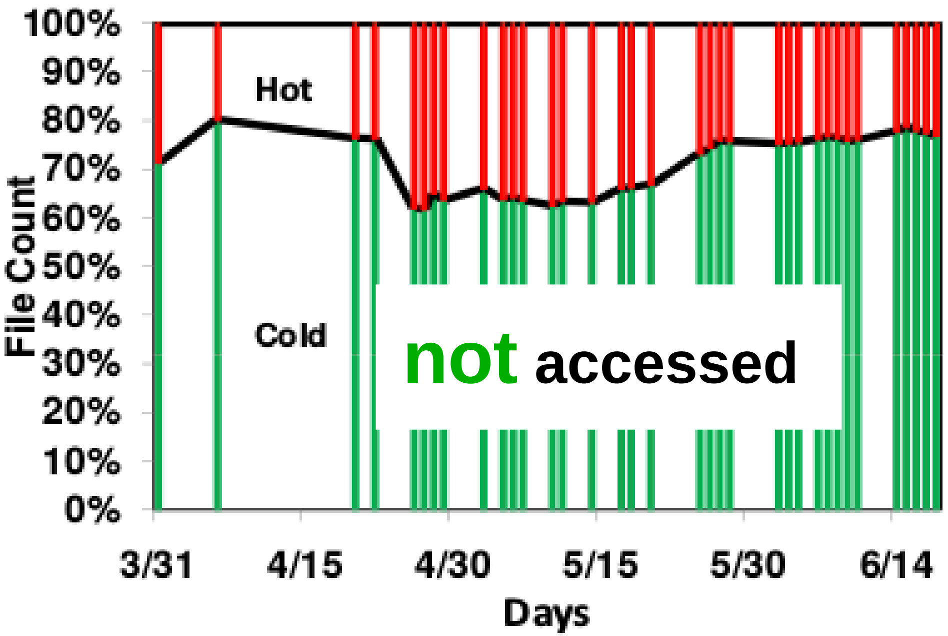
Joint work with
L. Ganesh, Y. Wang, L. Alvizi and M. Dahlin



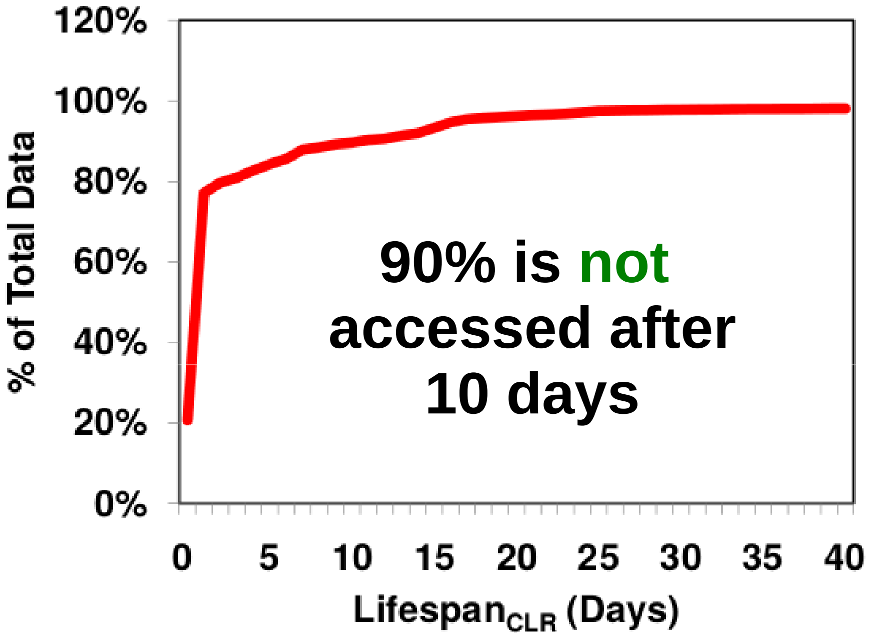
The amount of data is growing

BUT! lots of it is cold!

Cold data in Yahoo! cluster



not accessed



90% is not accessed after 10 days

From: “GreenHDFS: Towards An Energy-Conserving, Storage-Efficient, Hybrid Hadoop Compute Cluster”, Kaushik & Bhandnarkar, HotPower, 2010

Special properties of cold storage system

- Fast retrieval
- Few read accesses
- **Low cost**

vs. archival storage

vs. “hot” storage

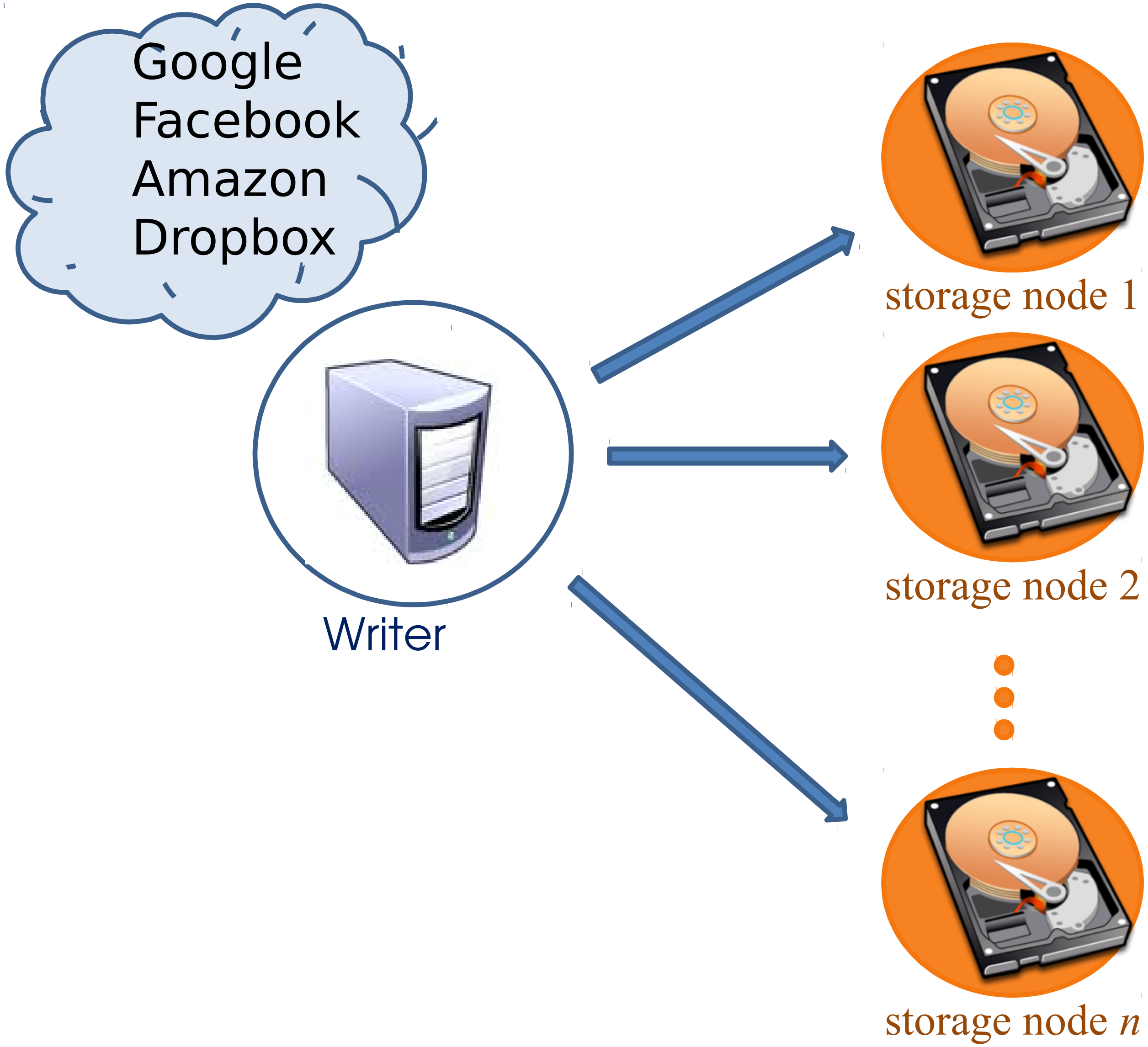
Cold = Existing users

Hot = New users

This talk

- Lazy recovery – a storage scheme for cold data
  - lower **network** cost
  - higher **storage** efficiency
  - higher **reliability**

Distributed storage system (DSS)



Google
Facebook
Amazon
Dropbox

Writer

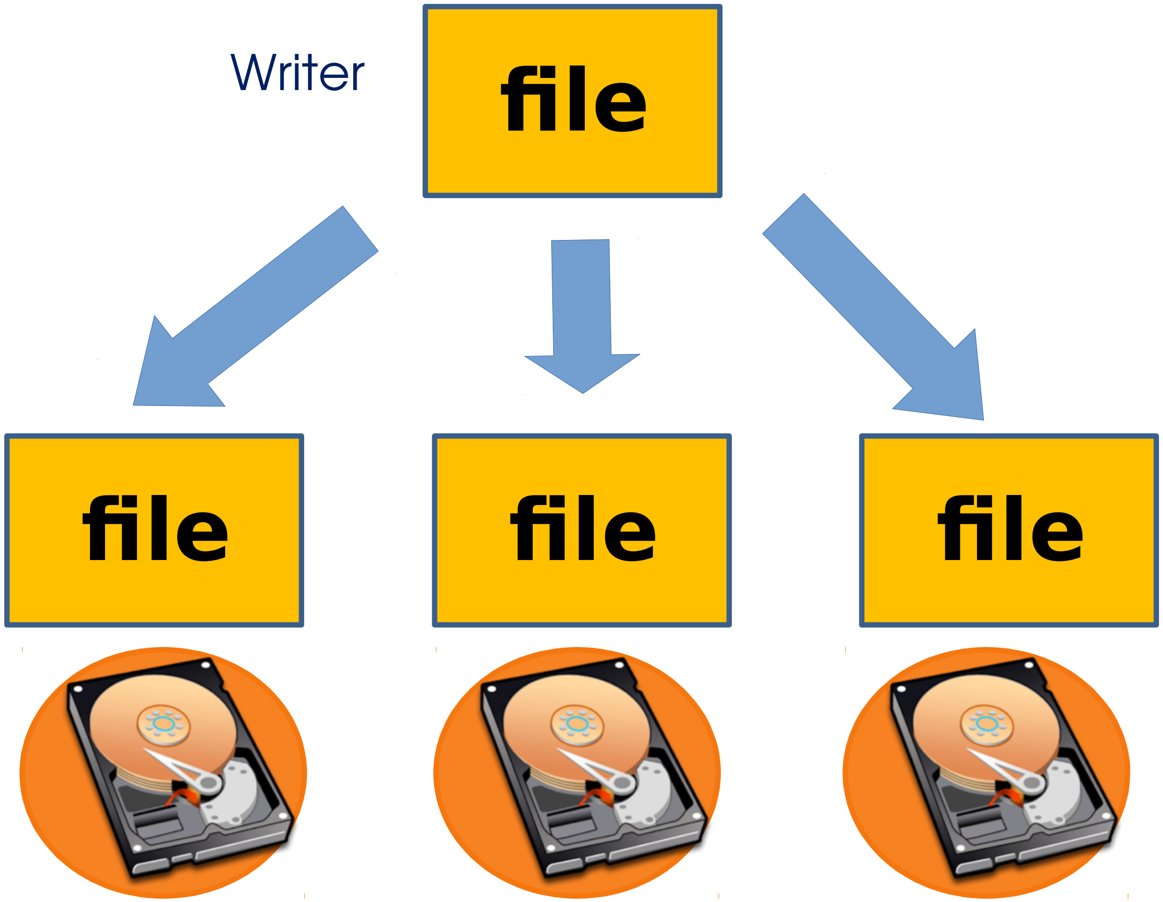
storage node 1

storage node 2

storage node *n*

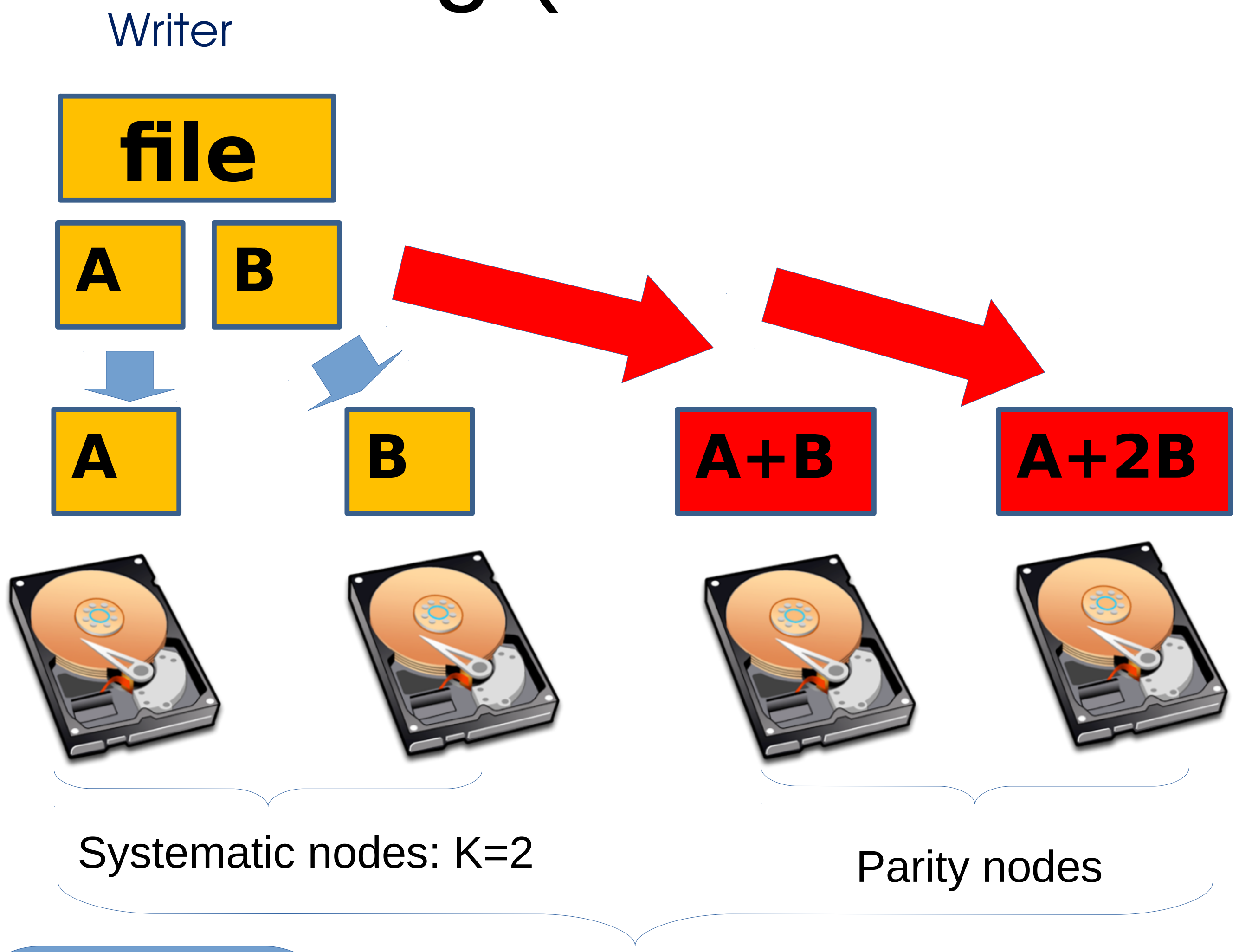
Need redundancy to handle failures

3x replication



Tolerates 2 node failures, pays **3x** in storage

Alternative to replication: Erasure coding (Reed-Solomon)



Writer

file

A B

A

B

A+B

A+2B

Systematic nodes: K=2

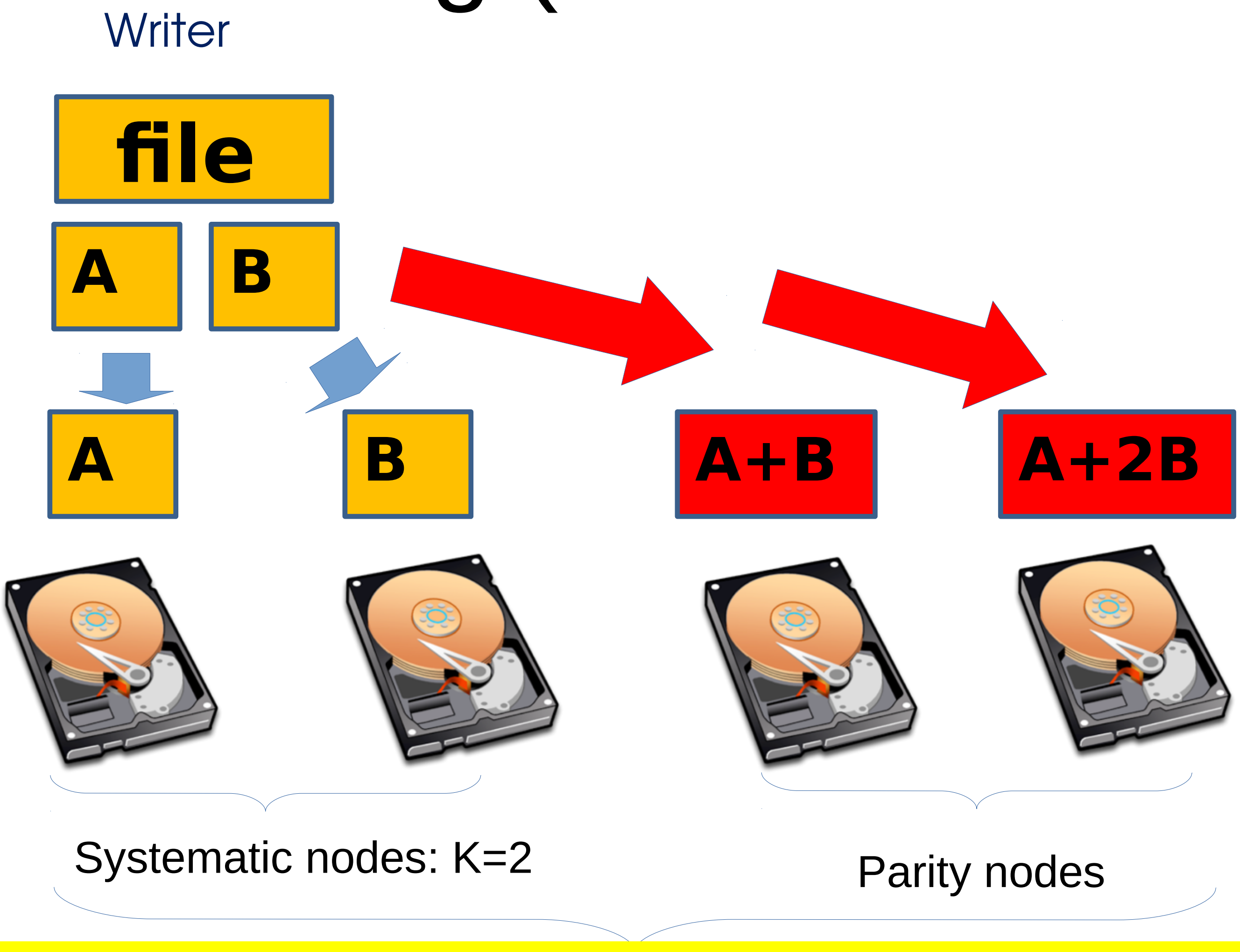
Parity nodes

Total nodes **per stripe**: N=4

RS(N=4,K=2)

N-K=2 → survives 2 failures

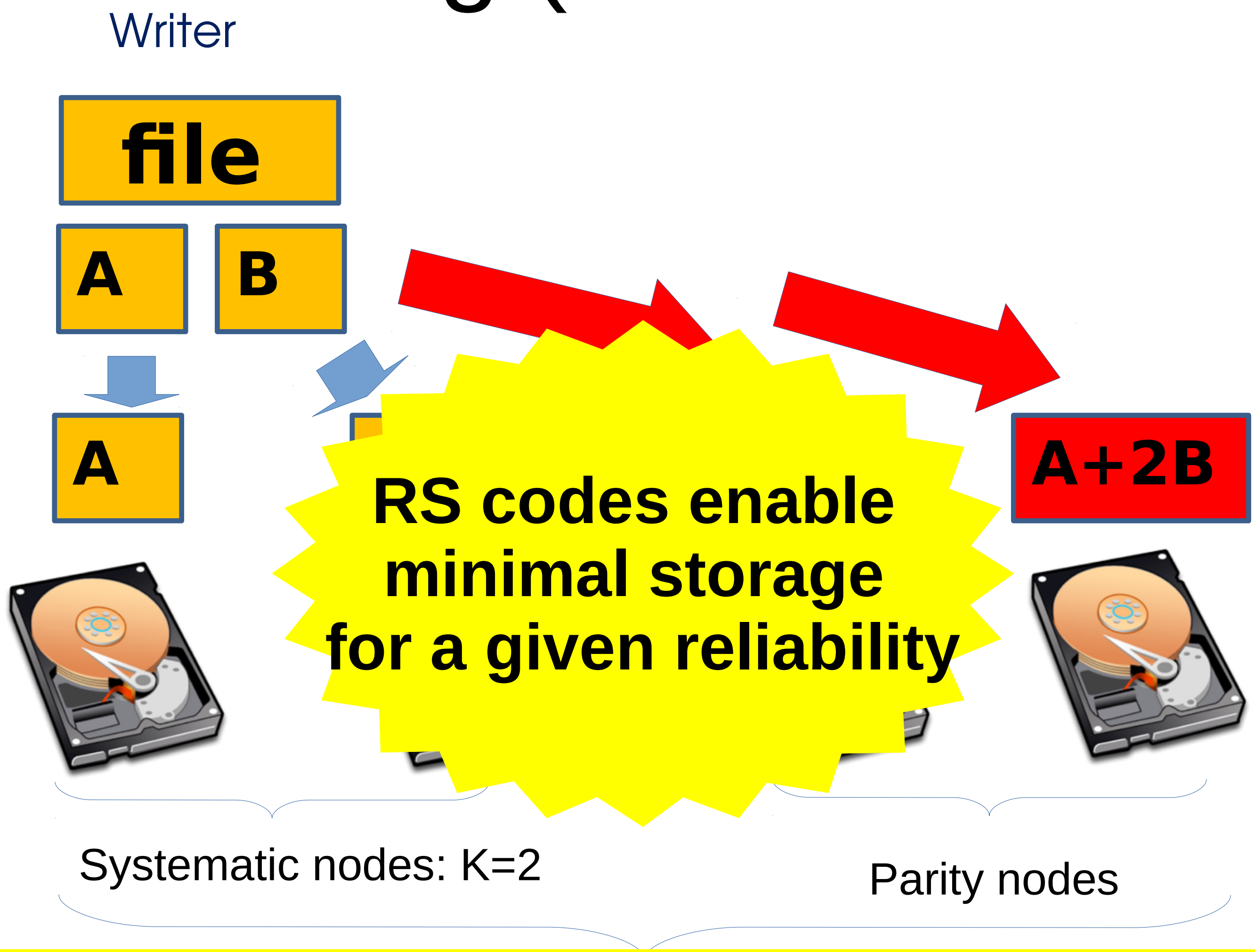
Alternative to replication: Erasure coding (Reed-Solomon)



Tolerates **2** node failure, pays **2x** in storage

2 failures

Alternative to replication: Erasure coding (Reed-Solomon)



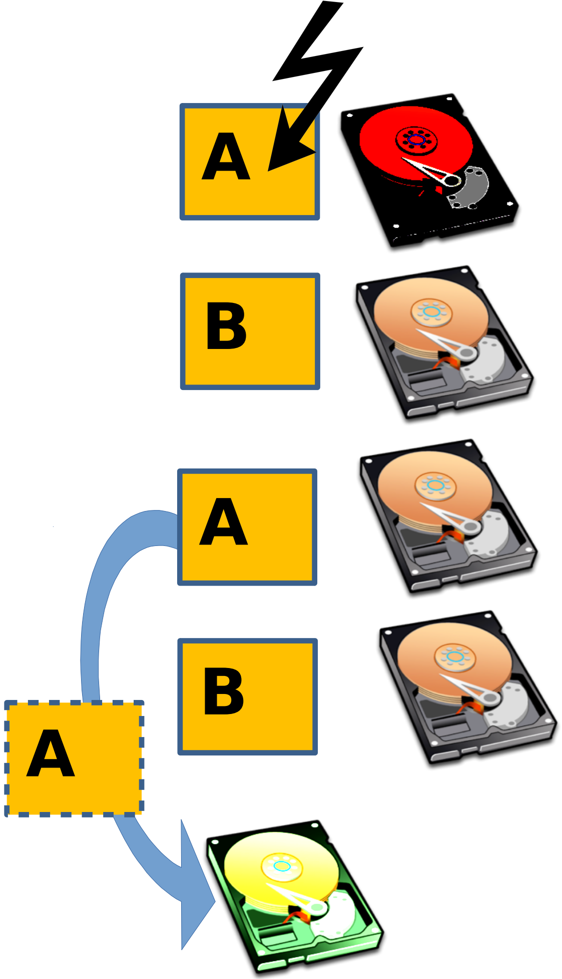
RS codes enable minimal storage for a given reliability

Tolerates **2** node failure, pays **2x** in storage

2 failures

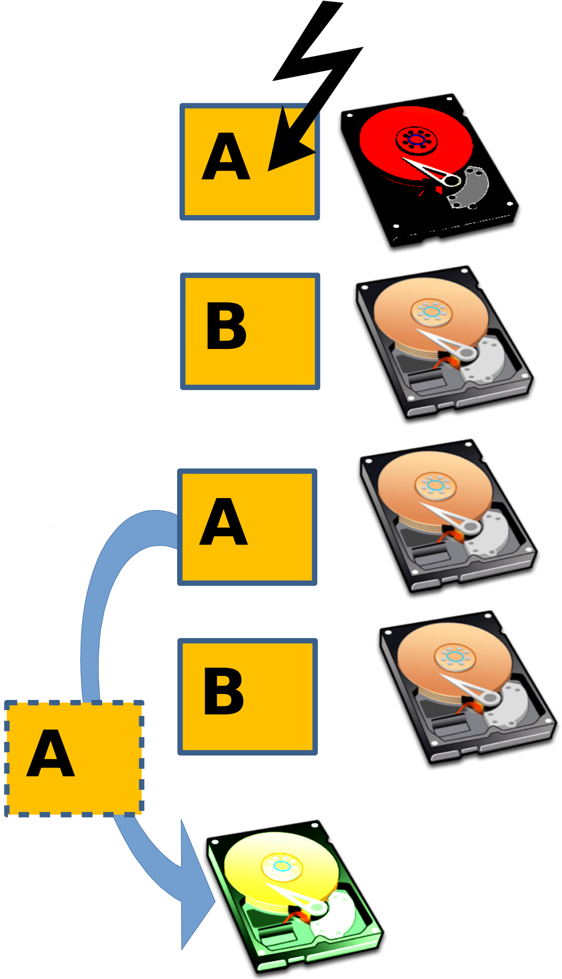
Repair bandwidth problem: recovery costs more bandwidth

Replication

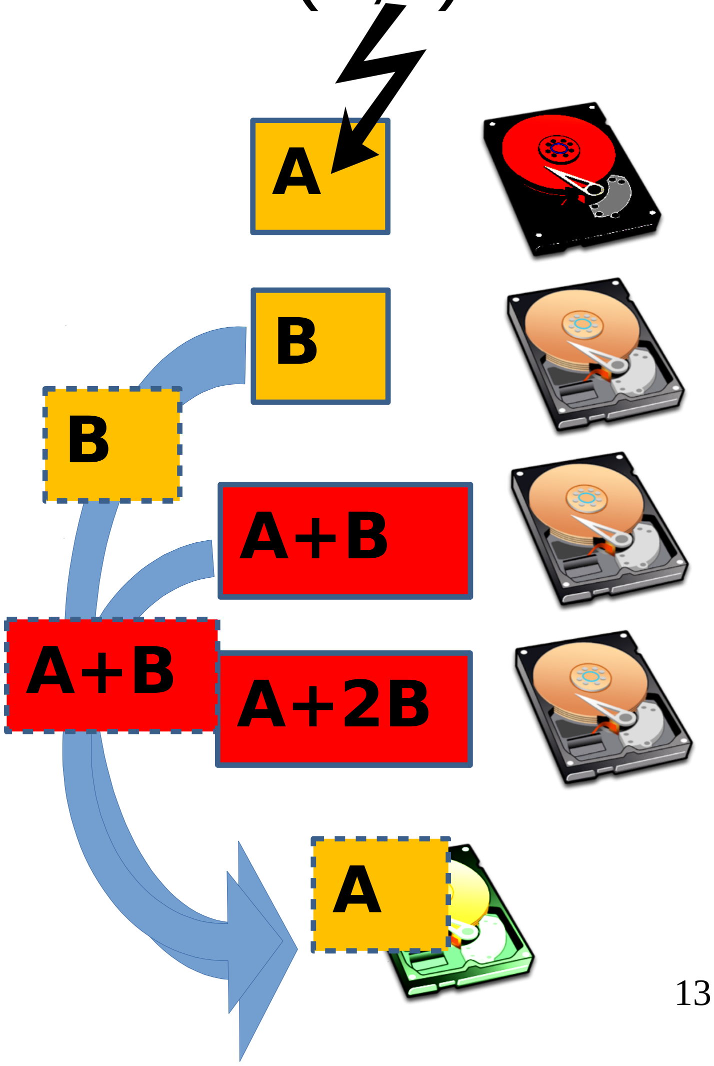


Repair bandwidth problem: recovery costs more bandwidth

Replication



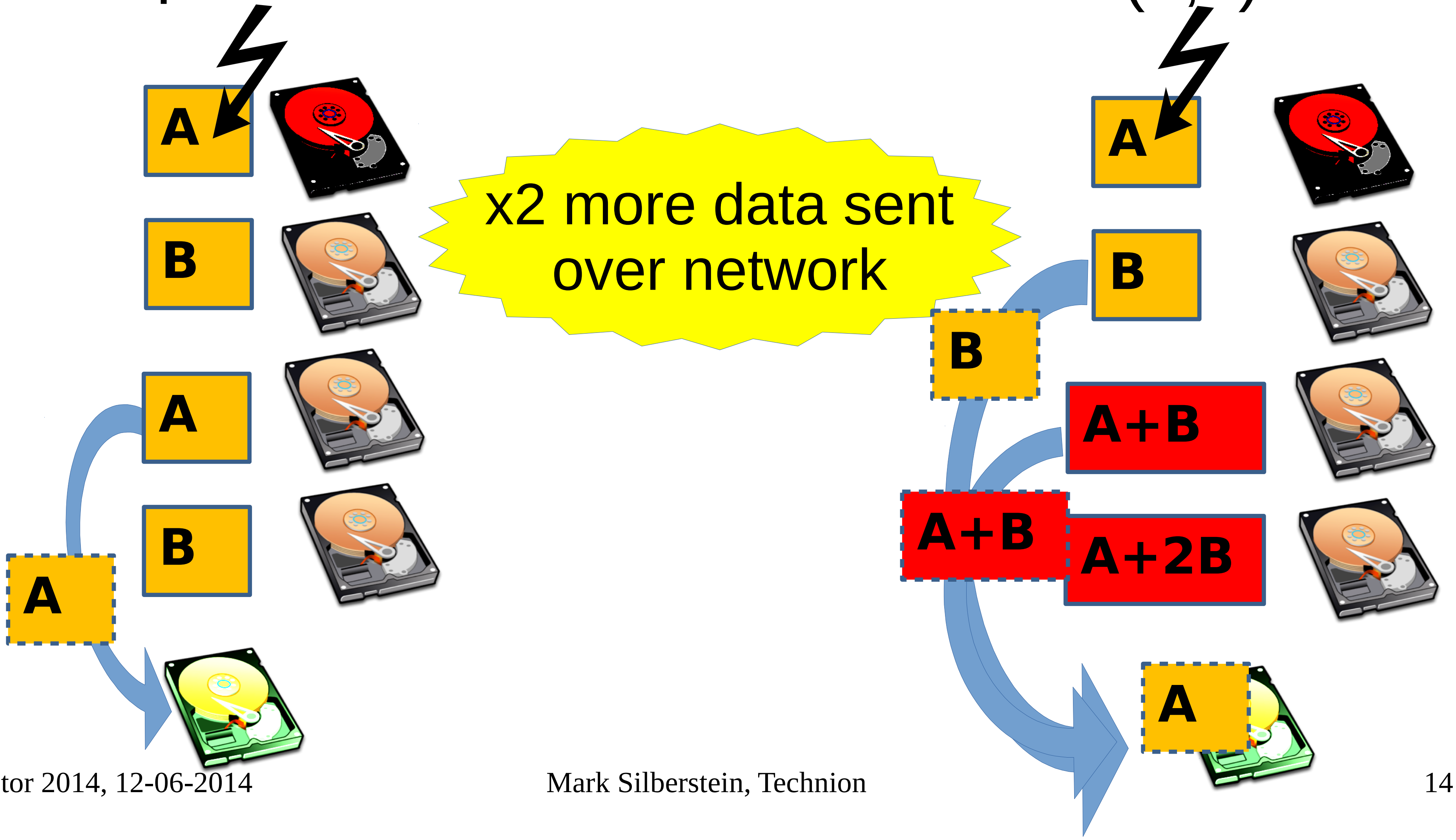
RS(4,2)



Repair bandwidth problem: recovery costs more bandwidth

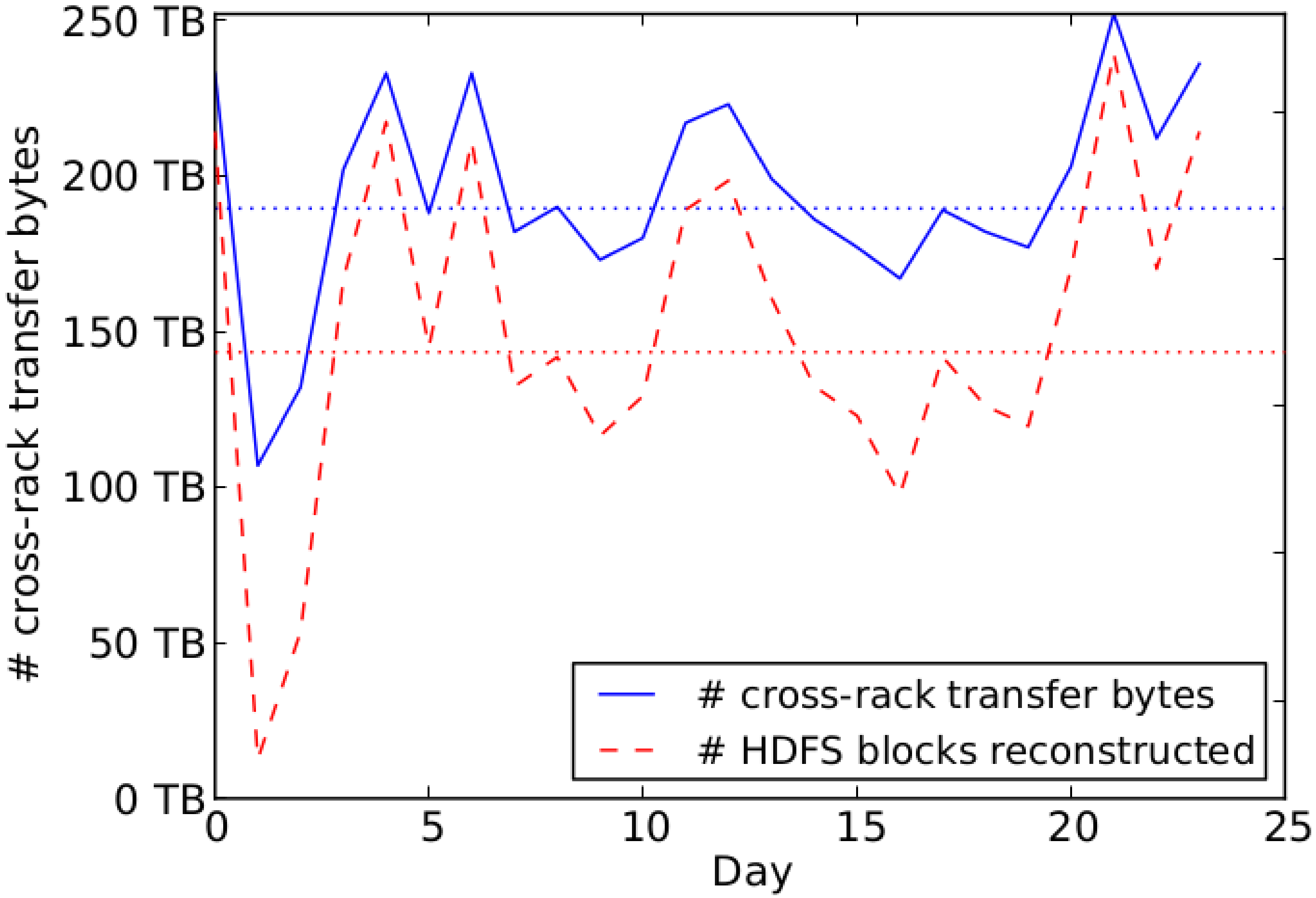
Replication

RS(4,2)



x2 more data sent over network

It is a real problem



Facebook N-nodes cluster, RS(14,10)

From K. Rashmi et.al., “A Solution to the Network Challenges of Data Recovery in Erasure-coded Distributed Storage Systems: A Study on the Facebook Warehouse Cluster”, HotStorage 2013

Root cause:
frequent recovery from **many nodes**

- Recovery is network-expensive!
- We pay the price after one node out of N failed

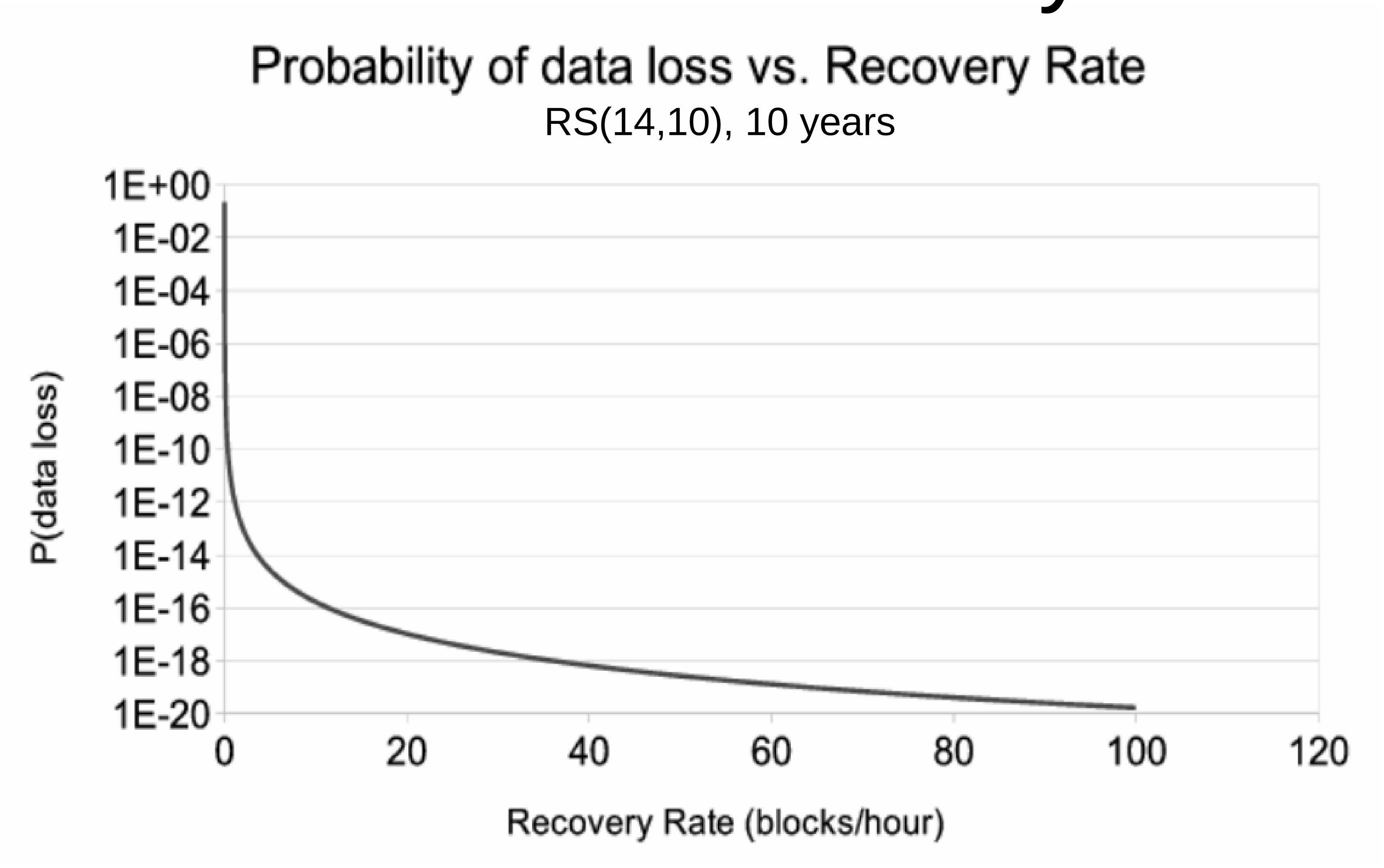
Root cause: **frequent** recovery from **many nodes**

Others: Change coding scheme to improve recovery costs

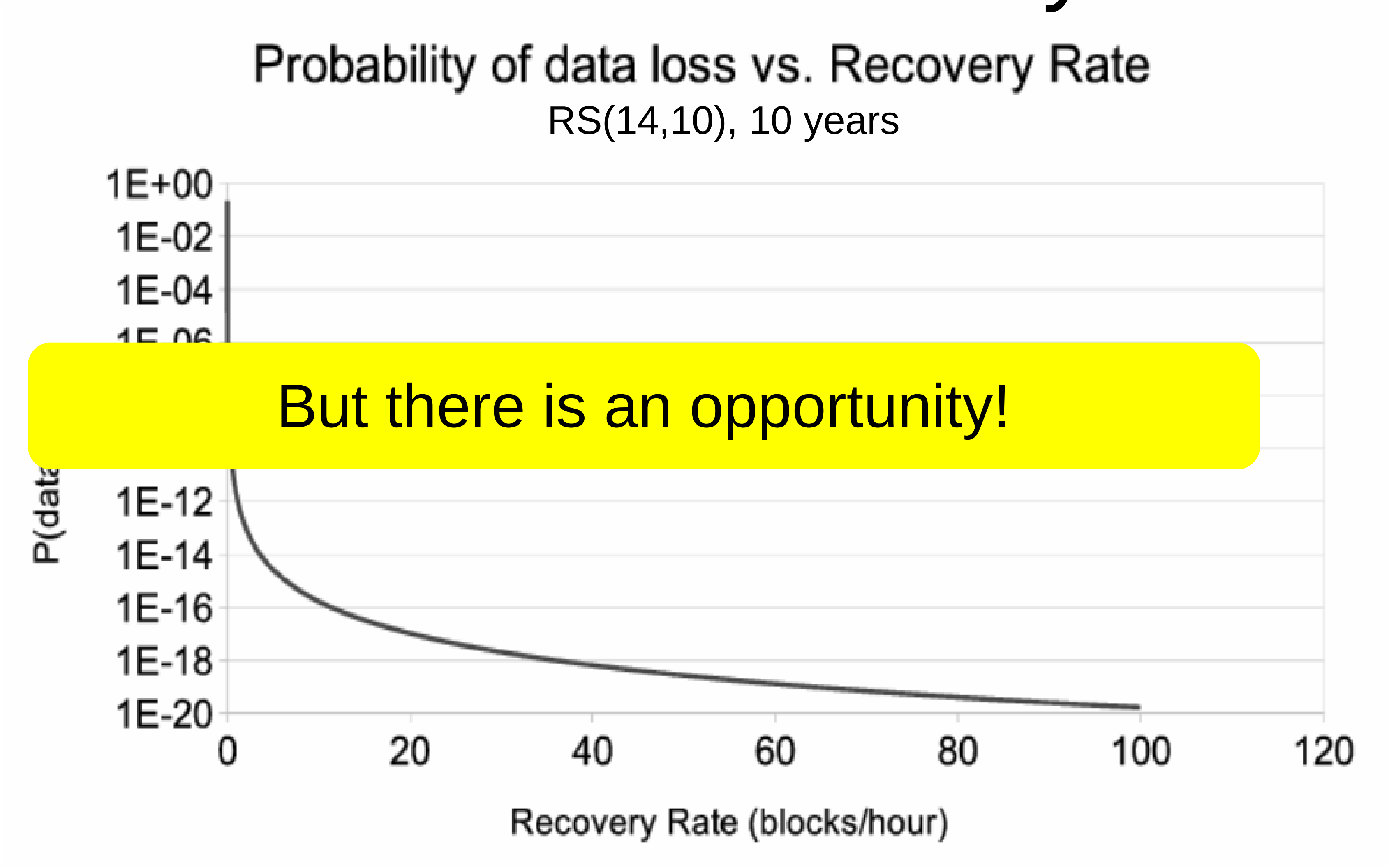
- Recovery is network-expensive!
- We pay the price after one node out of N failed

This work: play with recovery frequency

Problem: decreasing recovery rate decreases durability

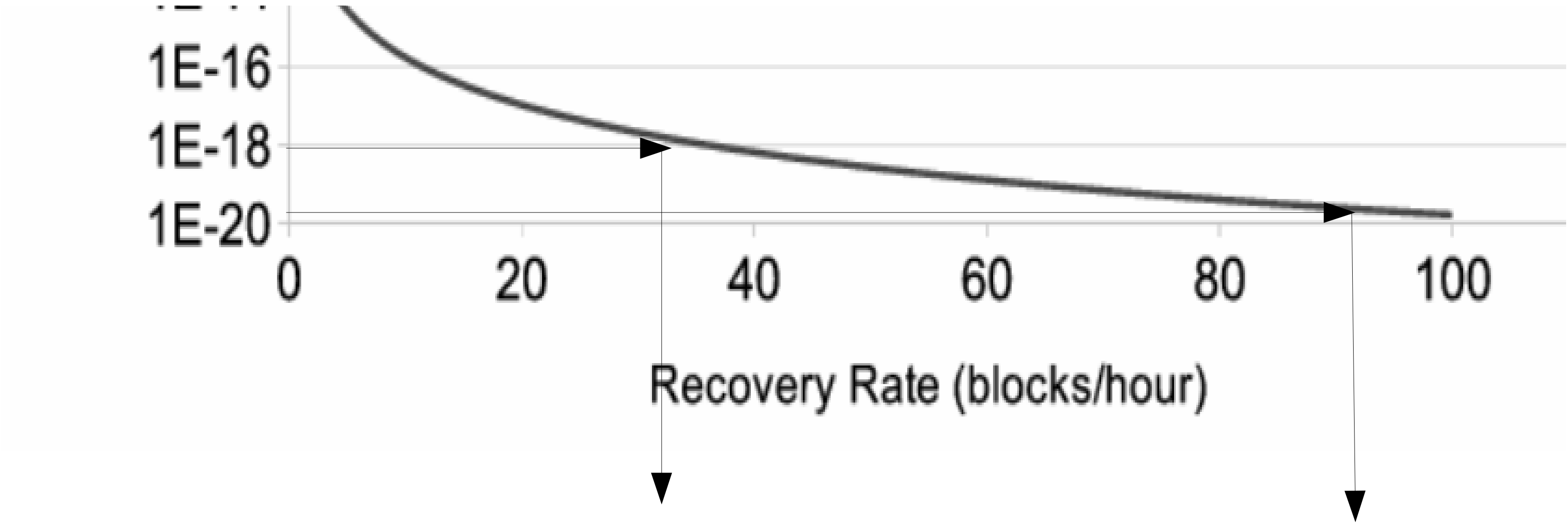


Problem: decreasing recovery rate decreases durability

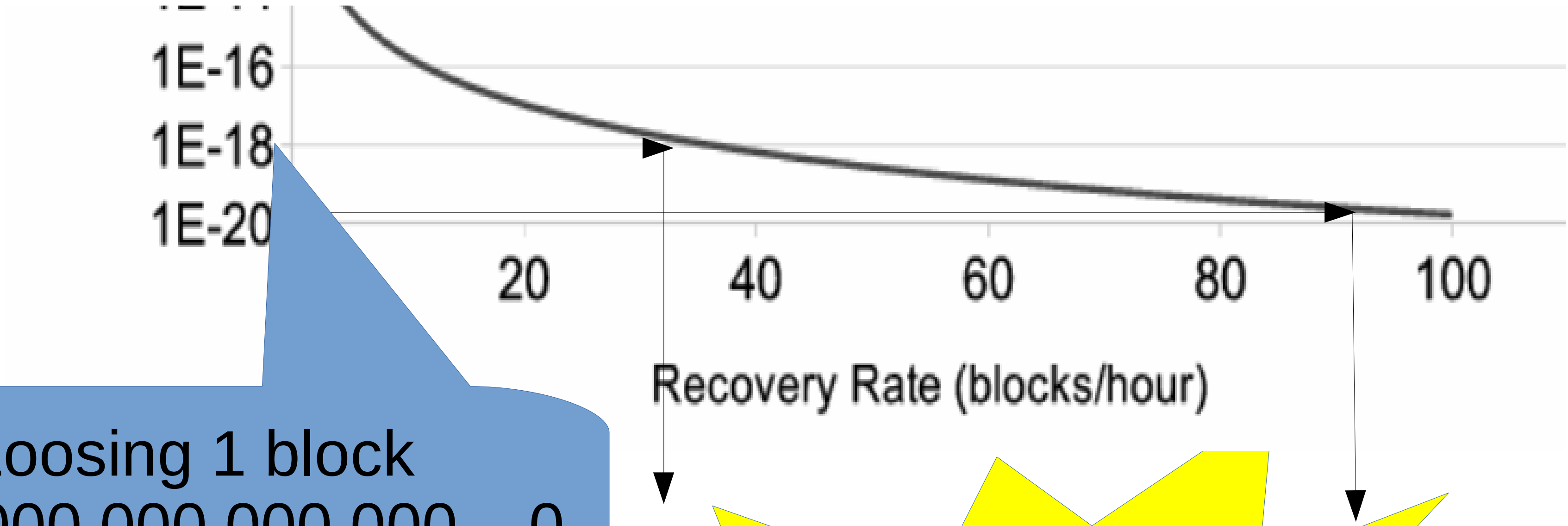


But there is an opportunity!

Do we really need durability that high?



Do we really need durability that high?



Loosing 1 block
in 10,000,000,000,000,...0
years

3x bandwidth
reduction

Lazy recovery approach

- Don't recover upon the first failure – wait until *F* failures
  - Was first used in P2P systems
- Benefits:
  - Less false-positive recoveries of transient failures
  - Recovery costs are amortized
- Slight decrease in reliability, slight increase in storage, massive decrease in bandwidth

3PB system RS(15,10)

Example: RS(15,10), recover 2 failures

Reliability: similar to RS(14,10)

	Standard (1 failure)	2 failures	3 failures
Repair traffic /day	**65 TB**	**15.3 TB**	**8 TB**

3PB system
RS(15,10)

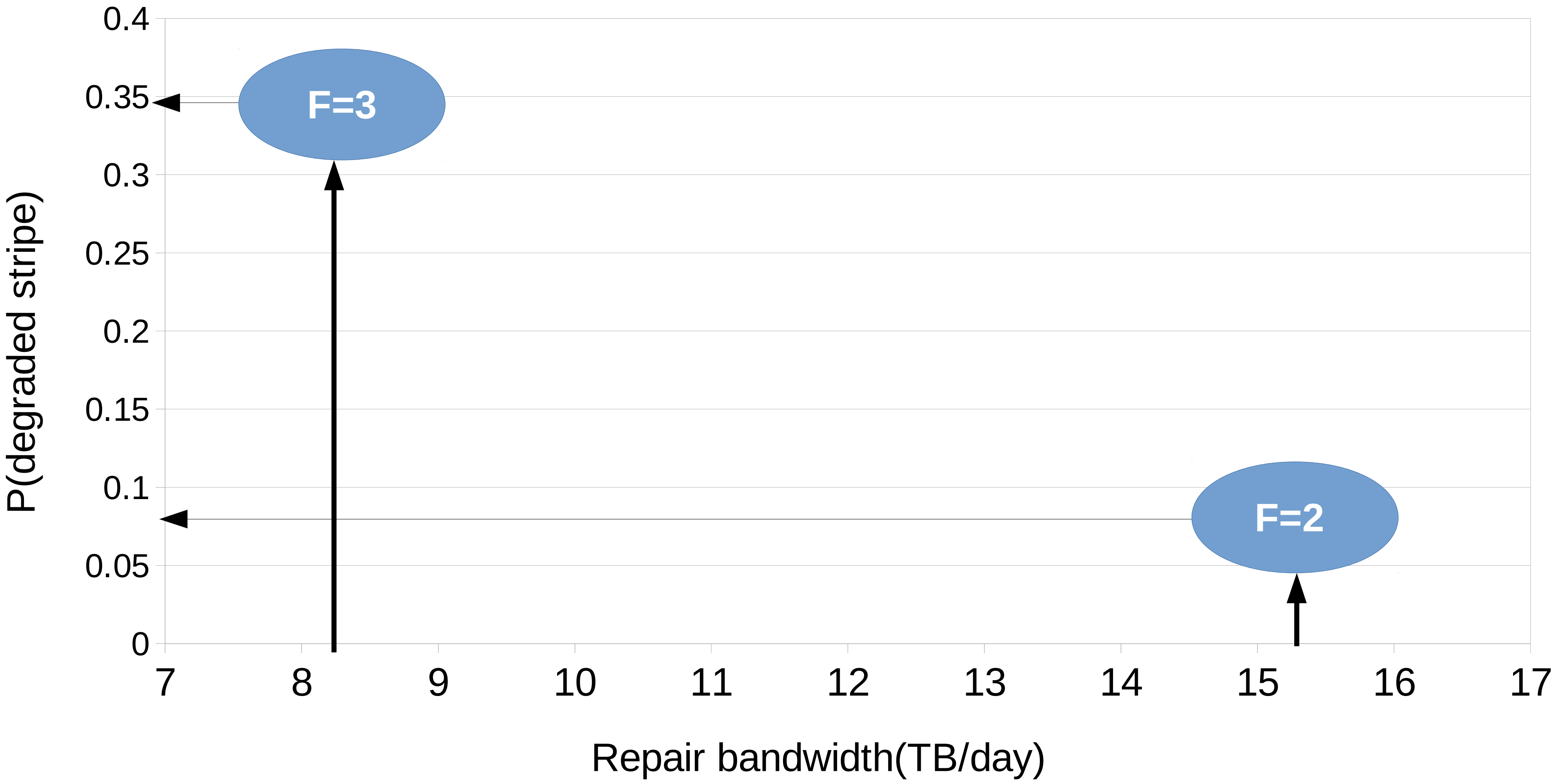
Example: RS(15,10), recover 2 failures

Reliability: similar to RS(14,10)

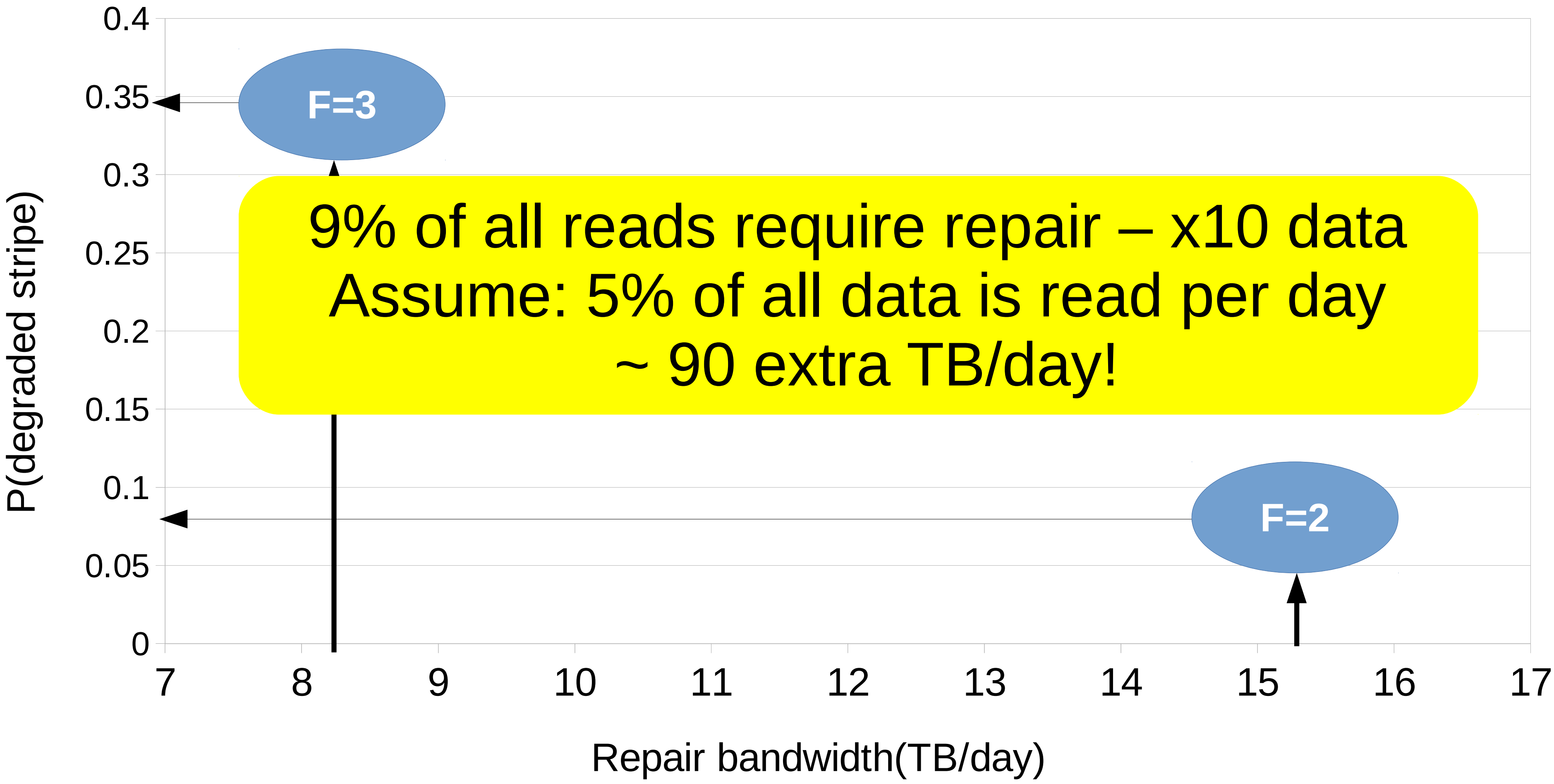
	Standard (1 failure)	2 failures	3 failures
Repair traffic /day	**65 TB**	**15.3 TB**	**8 TB**

BUT!!!

Lots of degraded stripes!

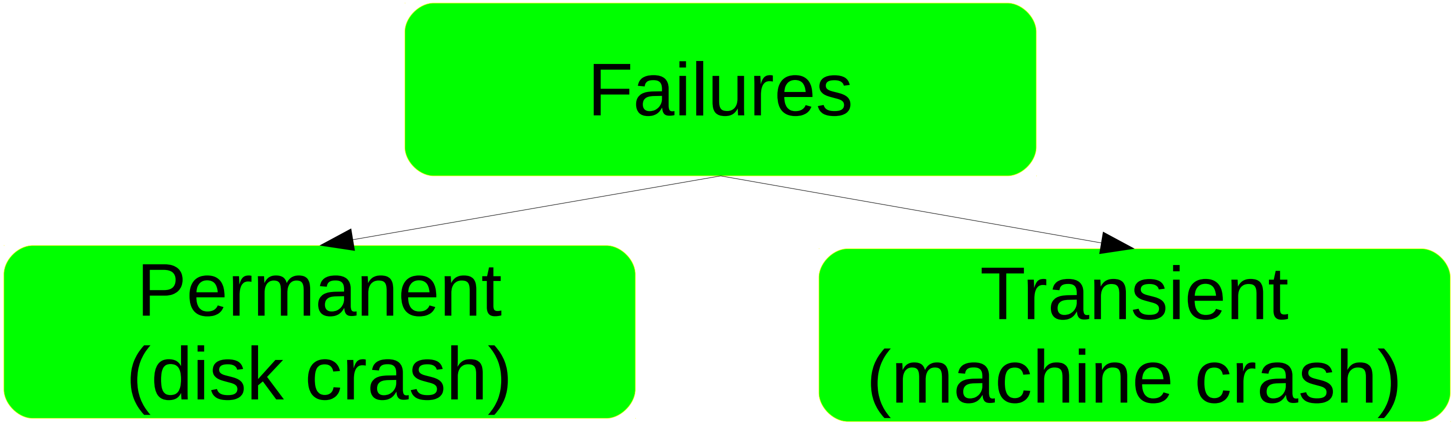


Lots of degraded stripes!



9% of all reads require repair – x10 data
Assume: 5% of all data is read per day
~ 90 extra TB/day!

Root cause

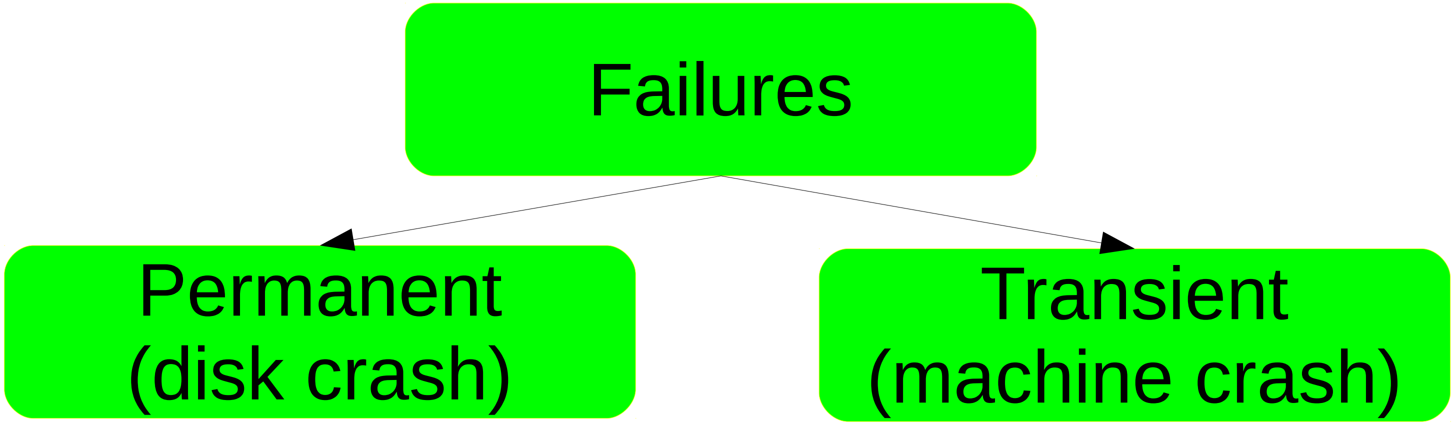

Failures

Permanent
(disk crash)

Transient
(machine crash)

- **Transient** recover by themselves – laziness pays off
- **Permanent** never recover – stripes remain degraded

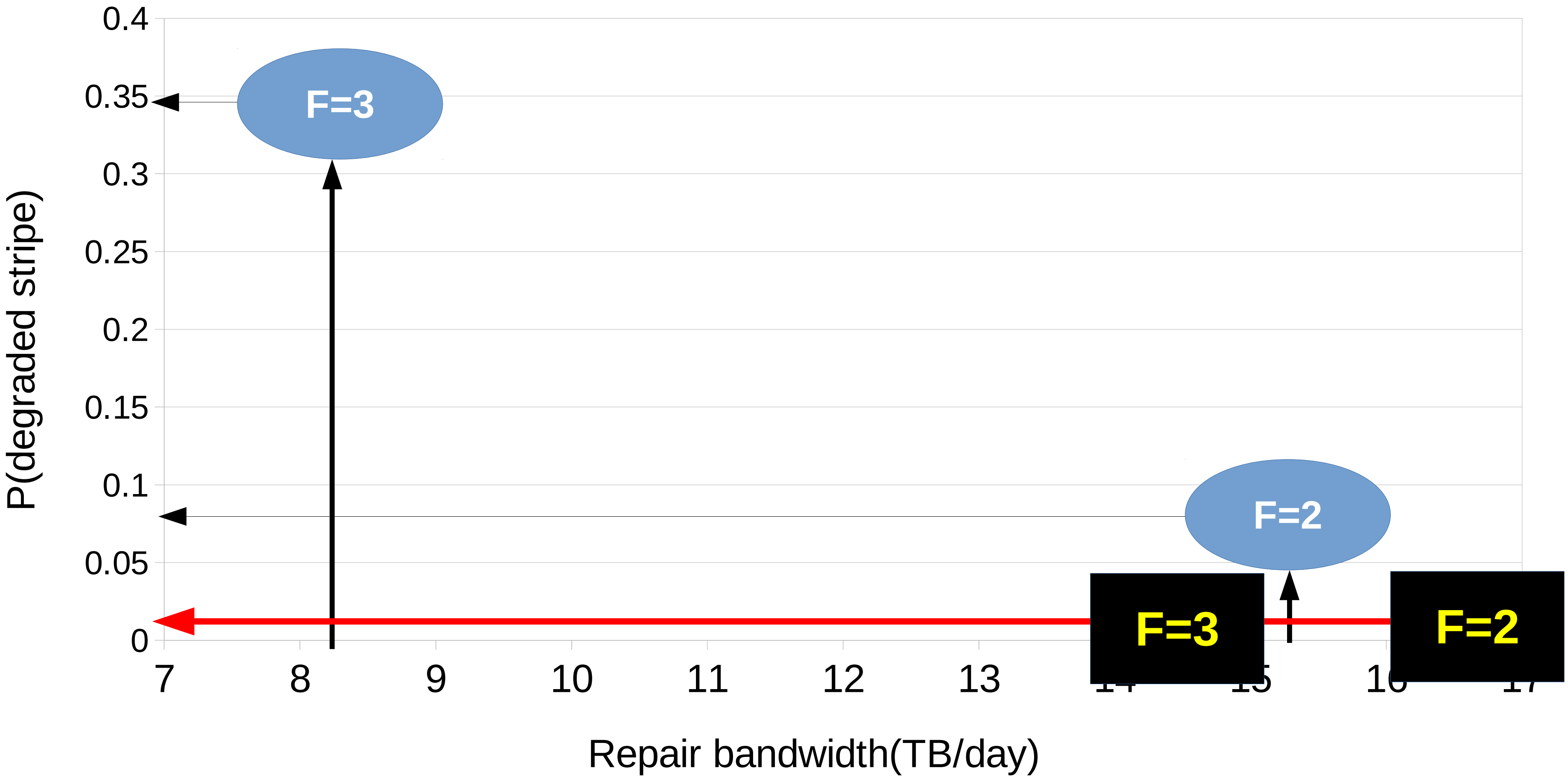
Root cause



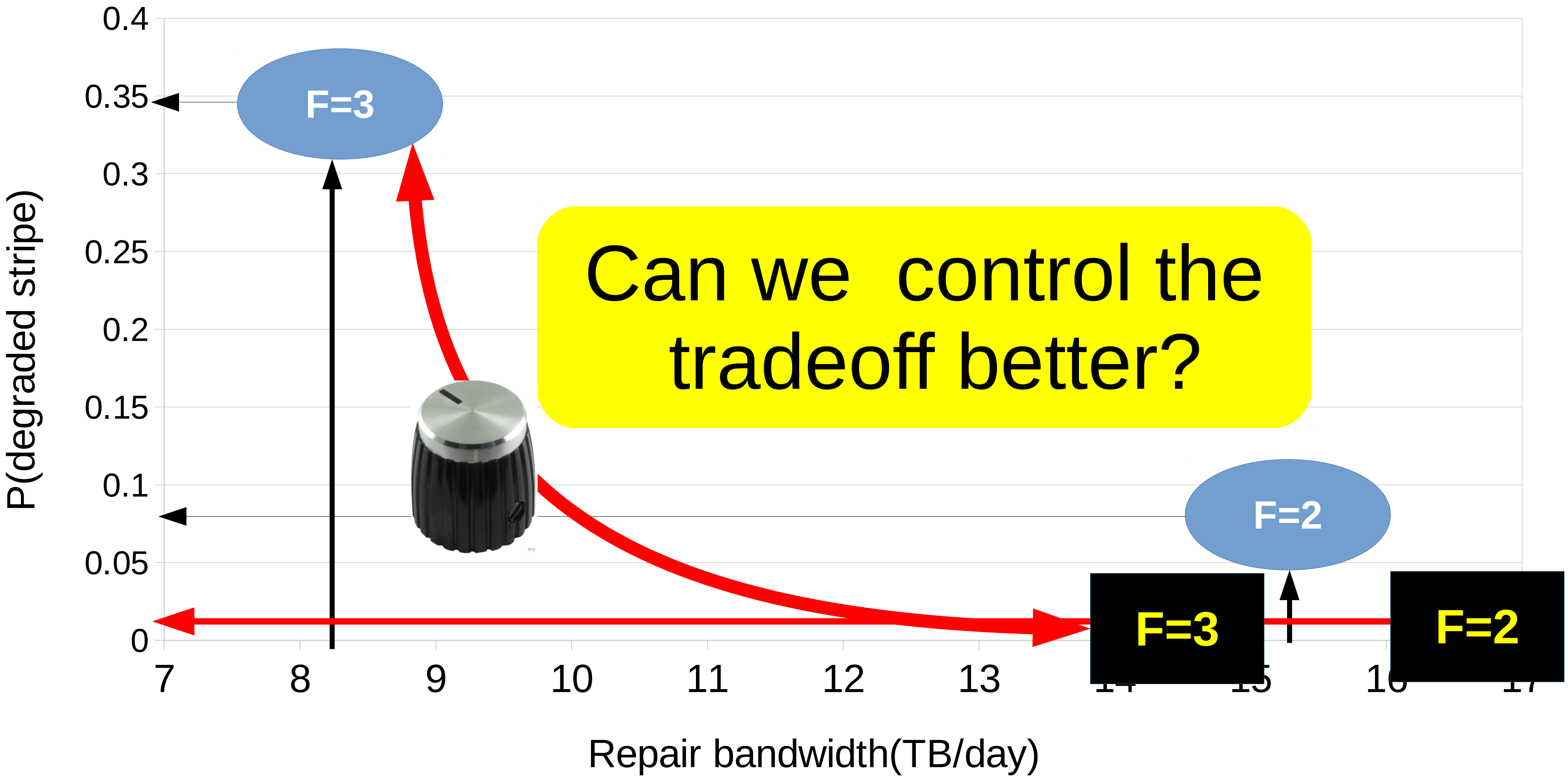
- **Transient** recover by themselves – laziness pays off
- **Permanent** never recover – stripes remain degraded

Try 2: use lazy recovery **ONLY** for transient failures

2% of degraded stripes



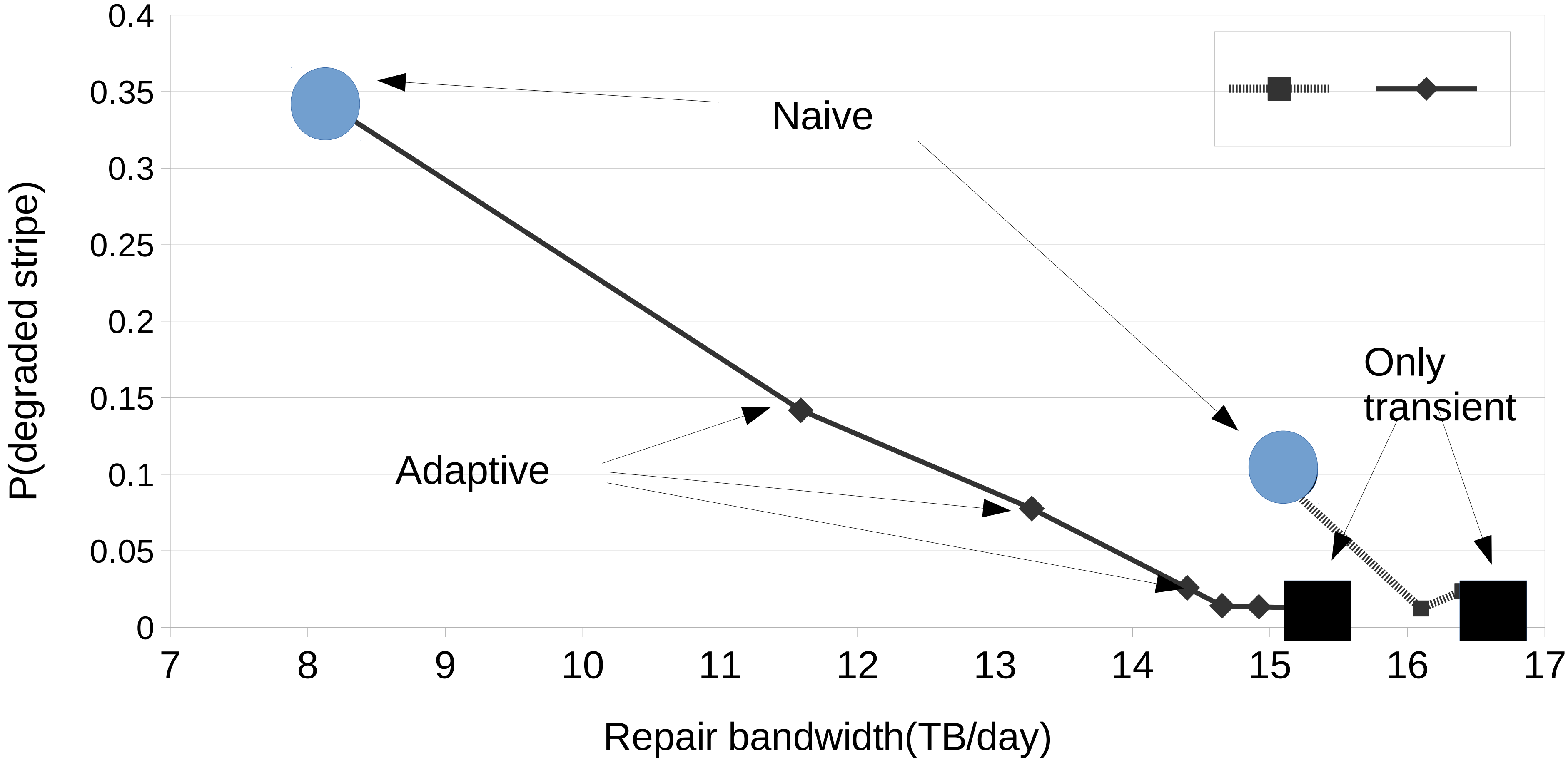
2% of degraded stripes



Adaptive recovery

- As before – lazy recovery for all transient failures
- Lazy recovery for permanent failures if system-wide permanently degraded below a target threshold
- Otherwise switch to eager recovery for permanent failures

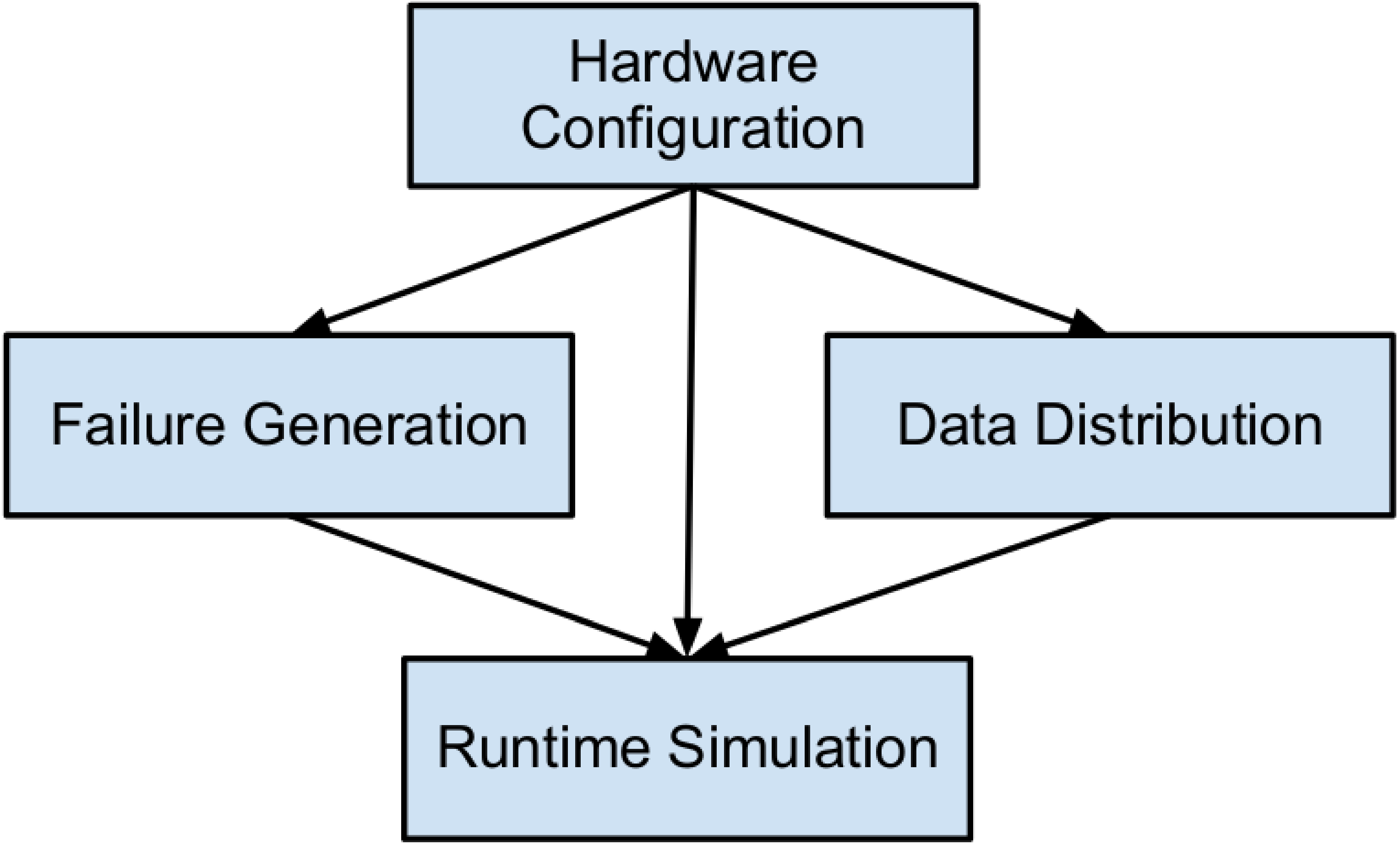
Putting it all together RS(15,10)



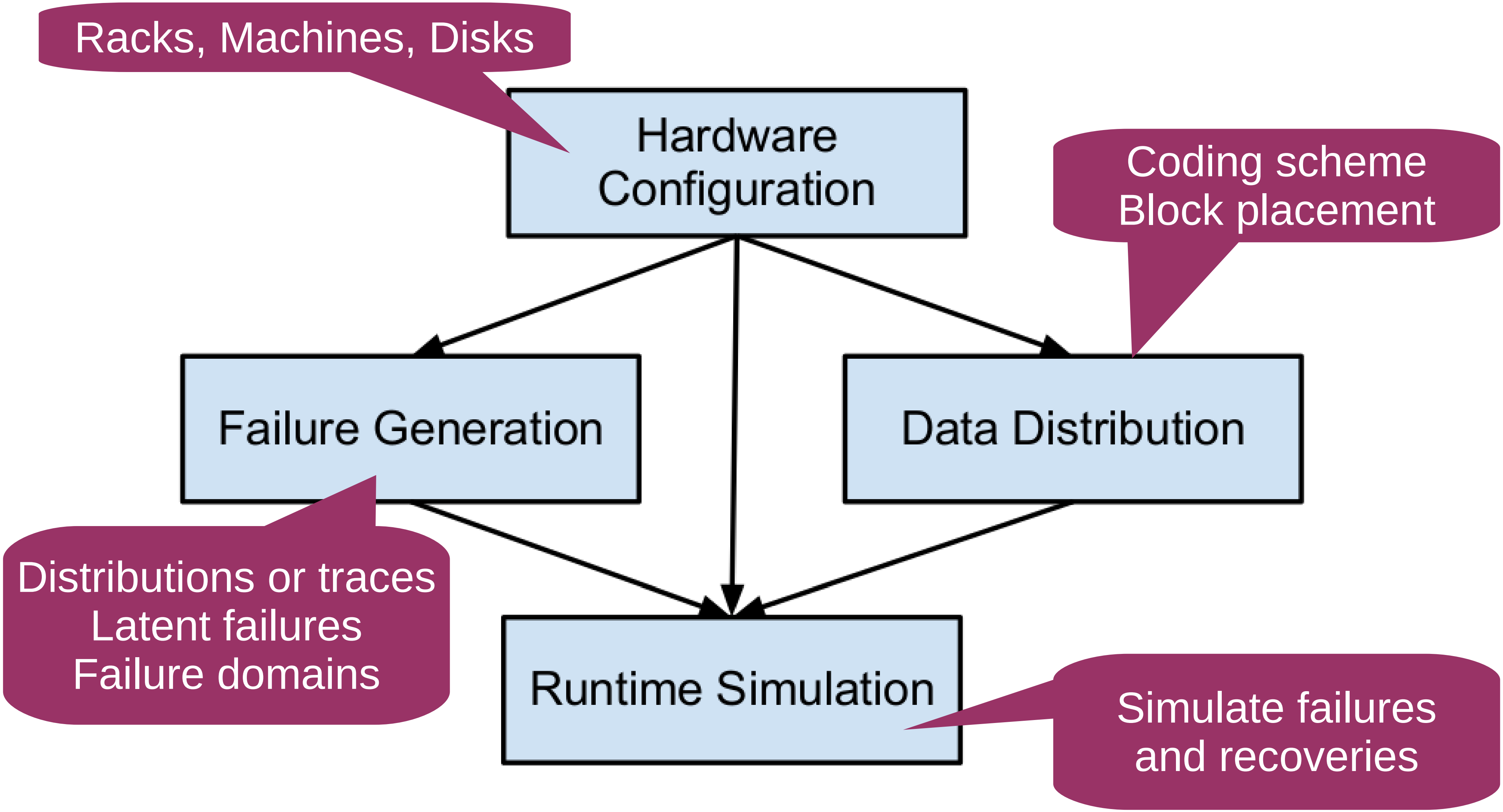
Evaluation methodology

- Problem: need to measure 10-years data-loss statistics of 3 PB system
- Solution:
  - Simulation: repair bandwidth and stripe degradation under realistic failure models
  - Modeling: data loss probability

DS-SIM: distributed storage simulator



DS-SIM: distributed storage simulator



Racks, Machines, Disks

Hardware Configuration

Coding scheme
Block placement

Failure Generation

Data Distribution

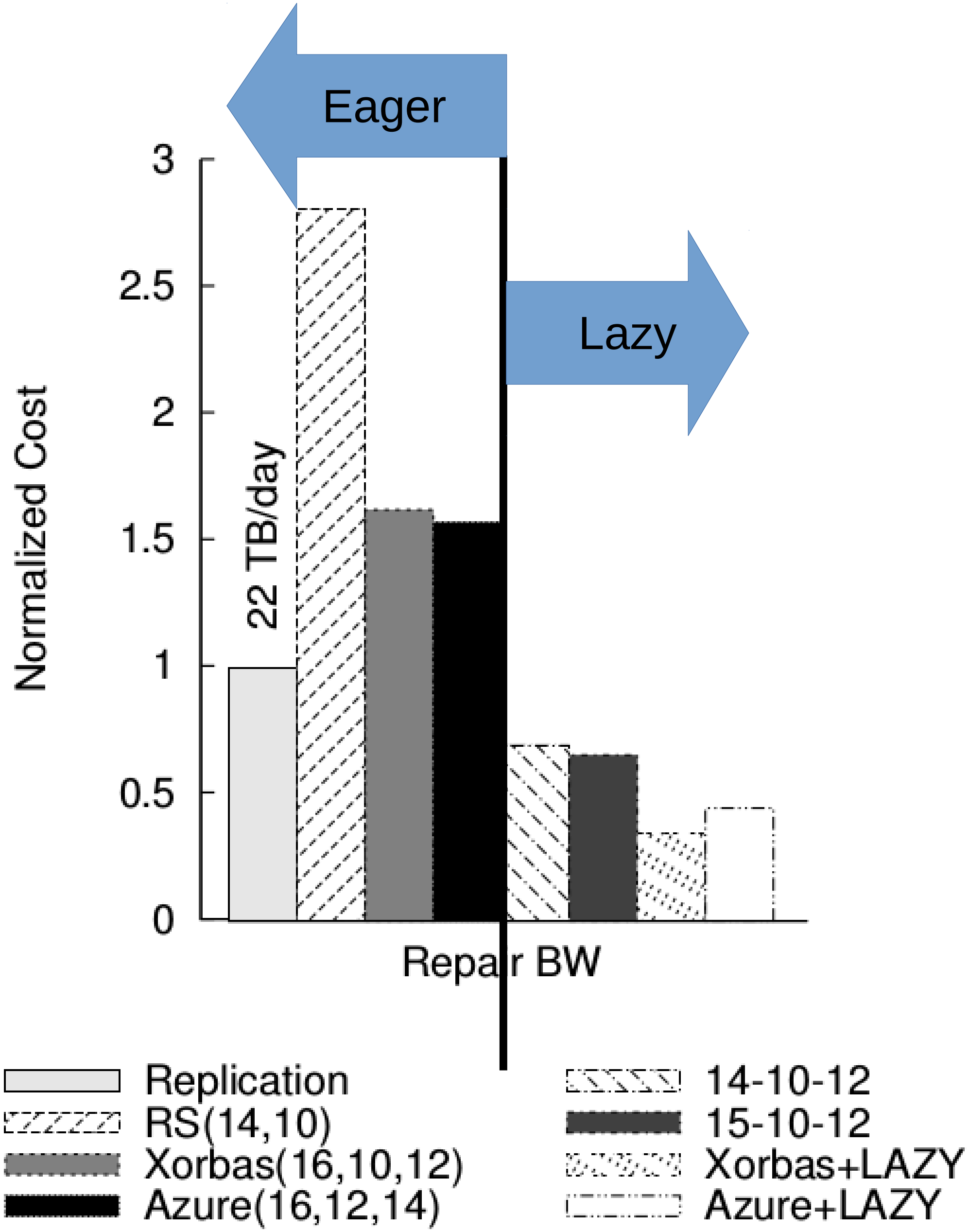
Distributions or traces
Latent failures
Failure domains

Runtime Simulation

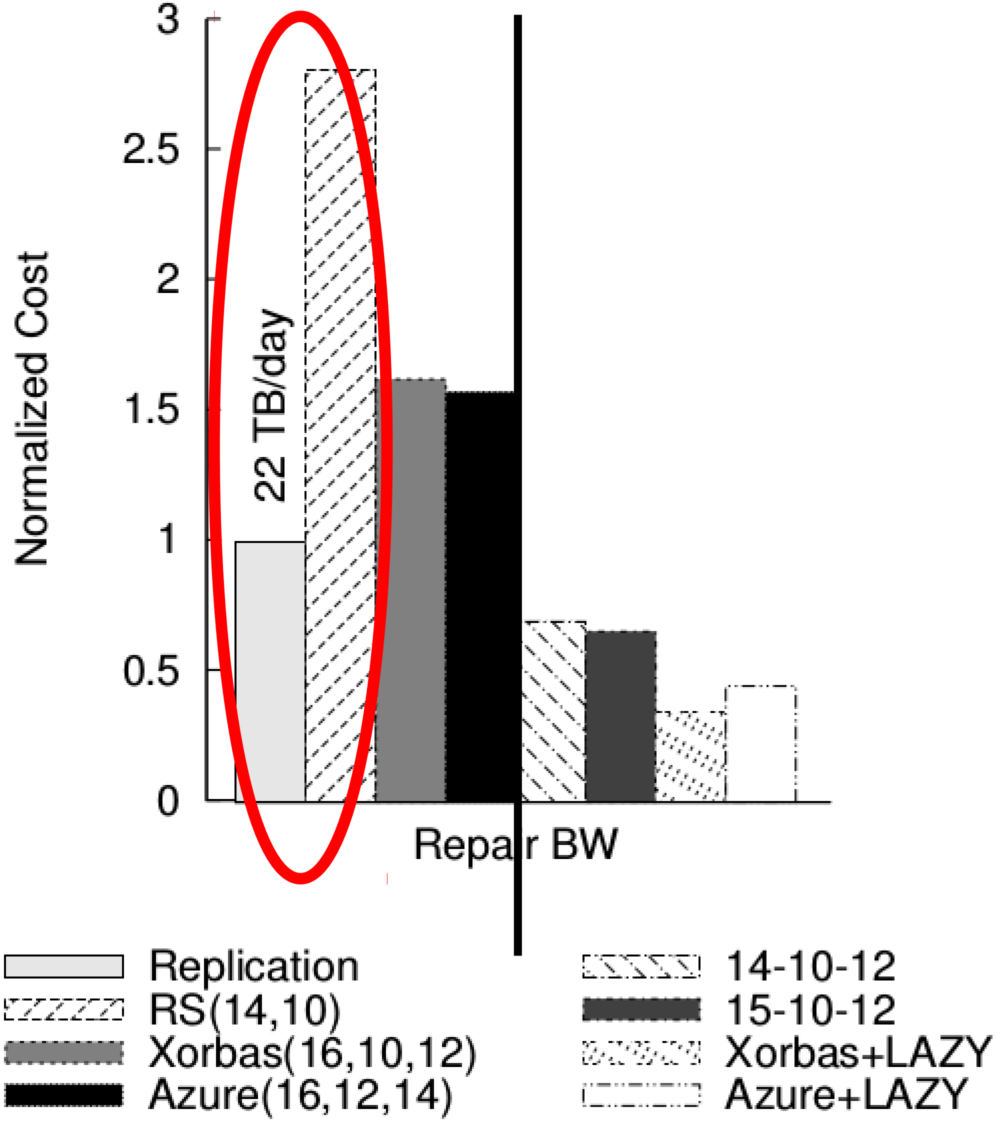
Simulate failures
and recoveries

Simulation parameters

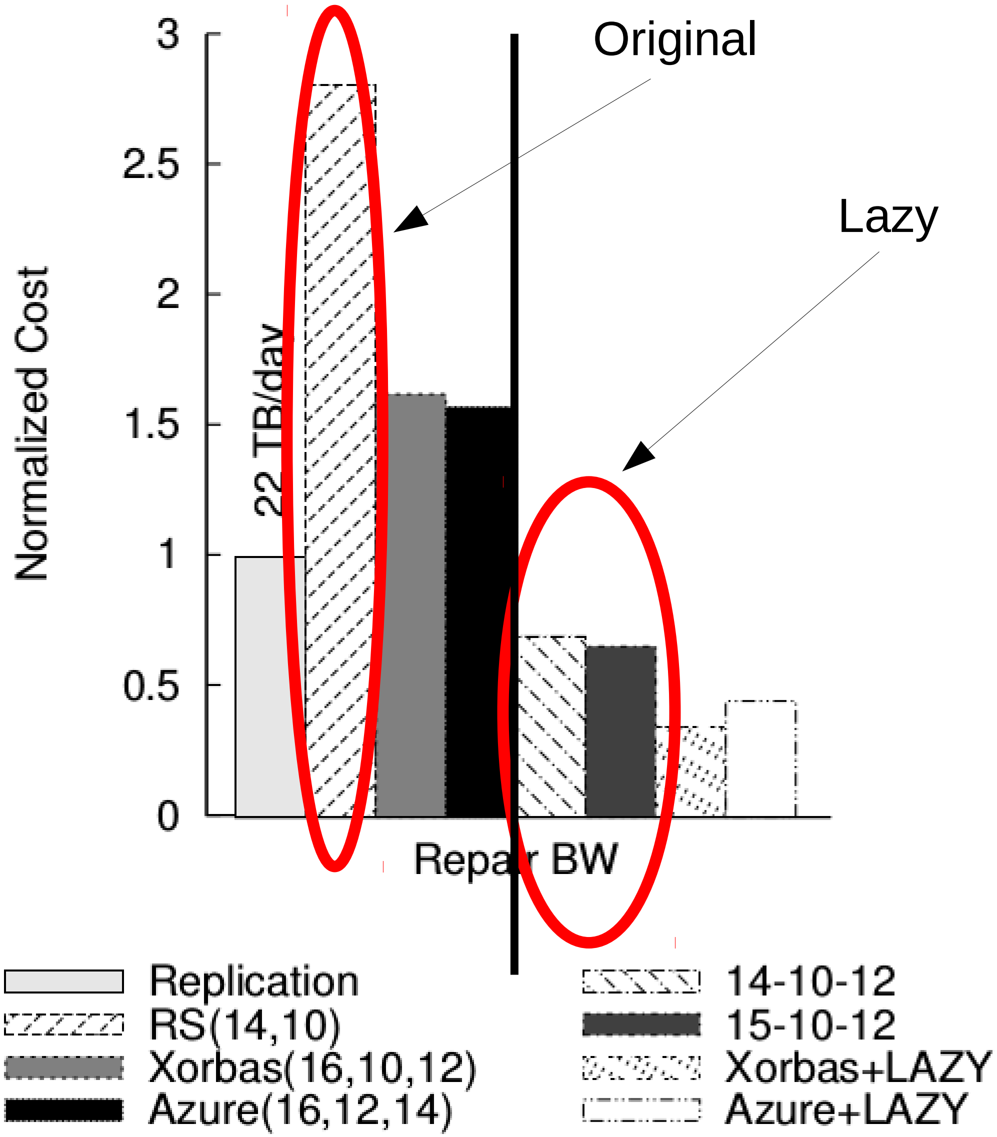
- 3 PB system, 35 racks, million strides, 10 years
- Failure distributions from previous works by Google, Facebook, Microsoft, Yahoo, CFDR trace repository
- 4 types of codes + their lazy versions


Eager
Lazy
3
2.5
2
1.5
1
0.5
0
Normalized Cost
22 TB/day
Repair BW
Replication
RS(14,10)
Xorbas(16,10,12)
Azure(16,12,14)
14-10-12
15-10-12
Xorbas+LAZY
Azure+LAZY

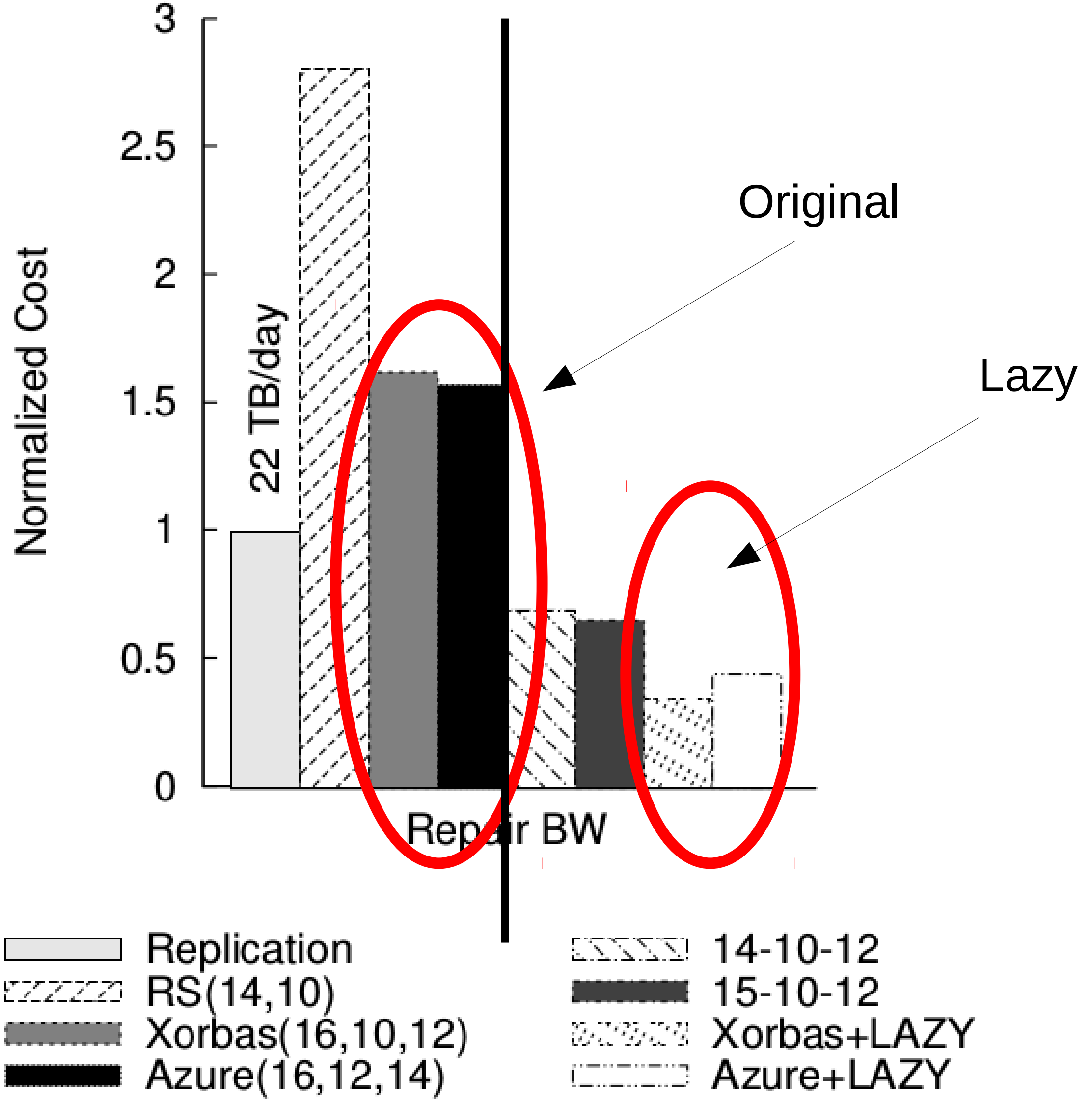
RS codes need ~3x repair bandwidth of replication



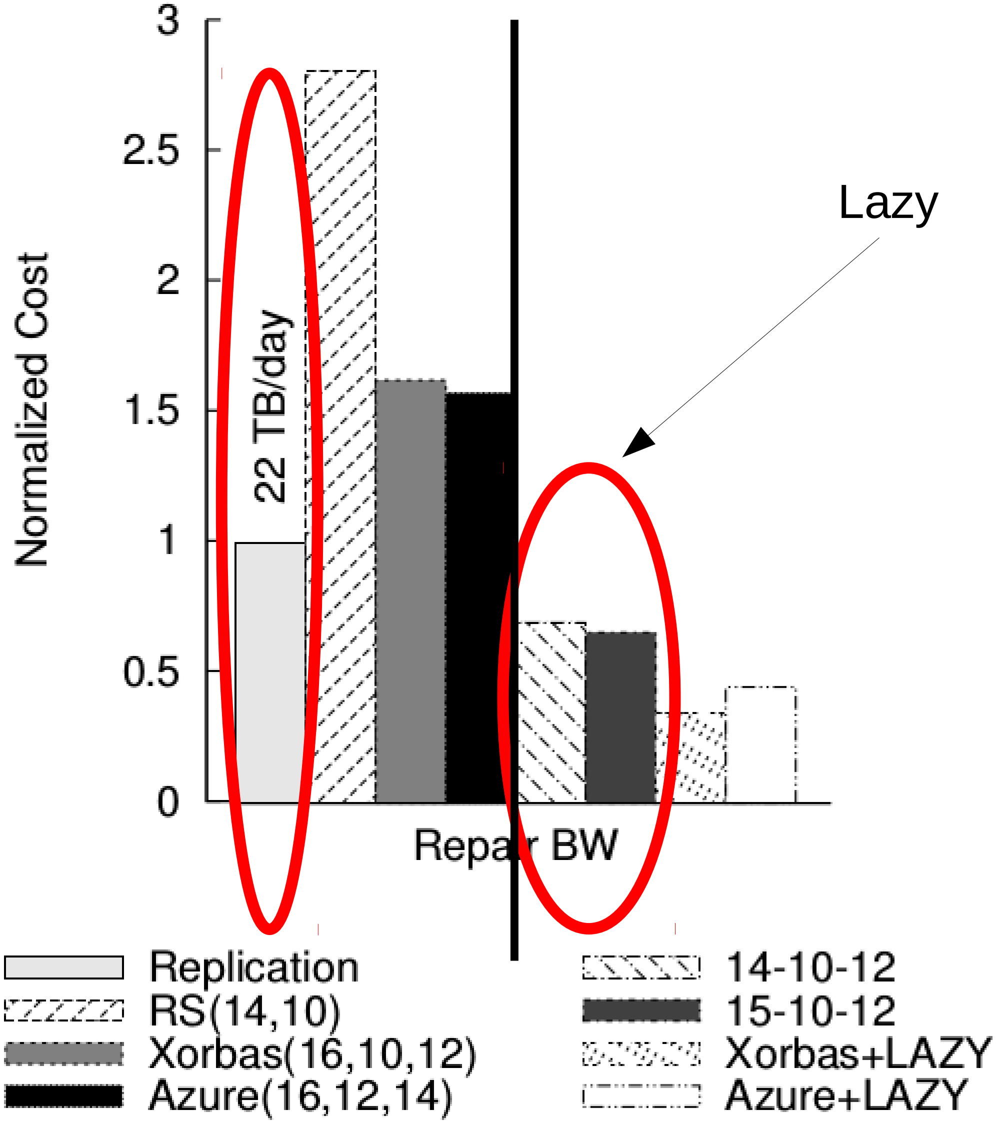
Lazy recovery bandwidth ~4x over traditional RS



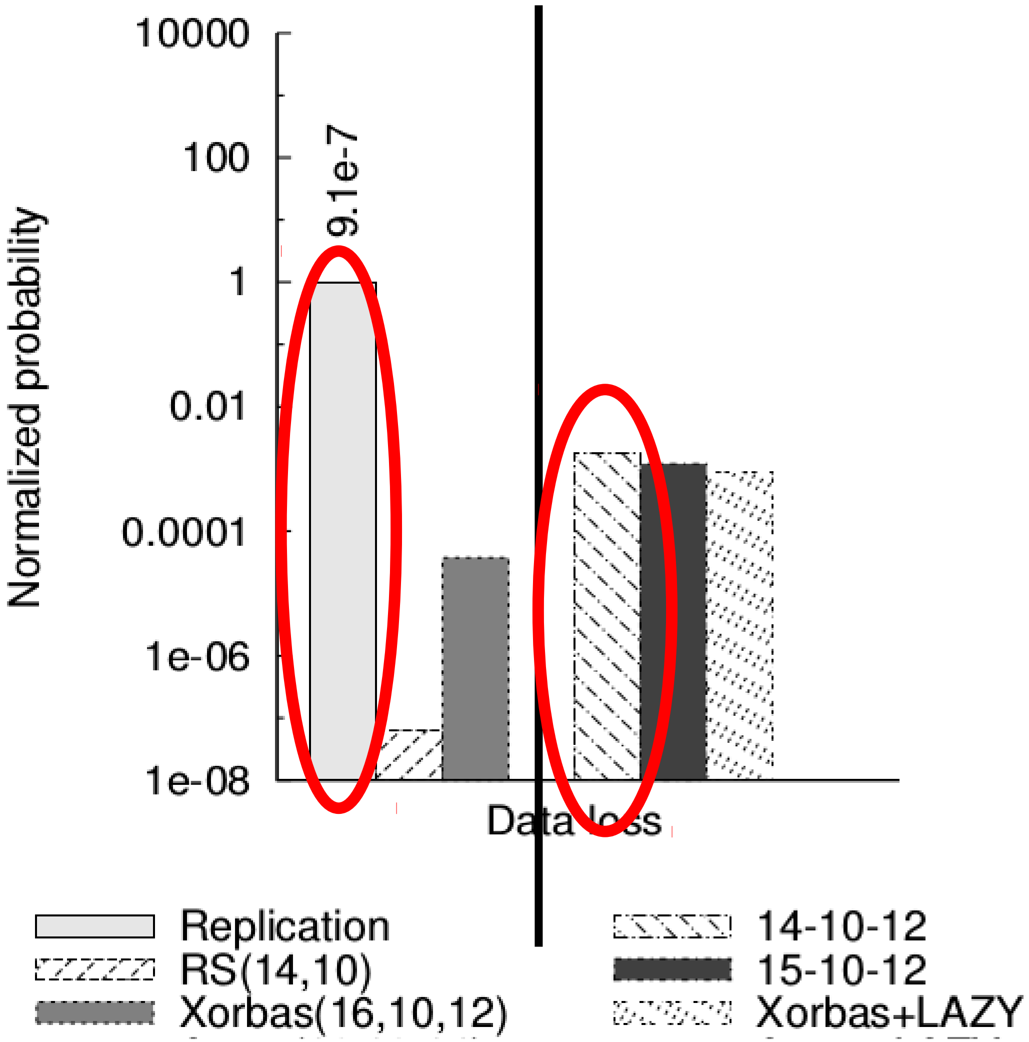
Lazy recovery improves repair-efficient codes



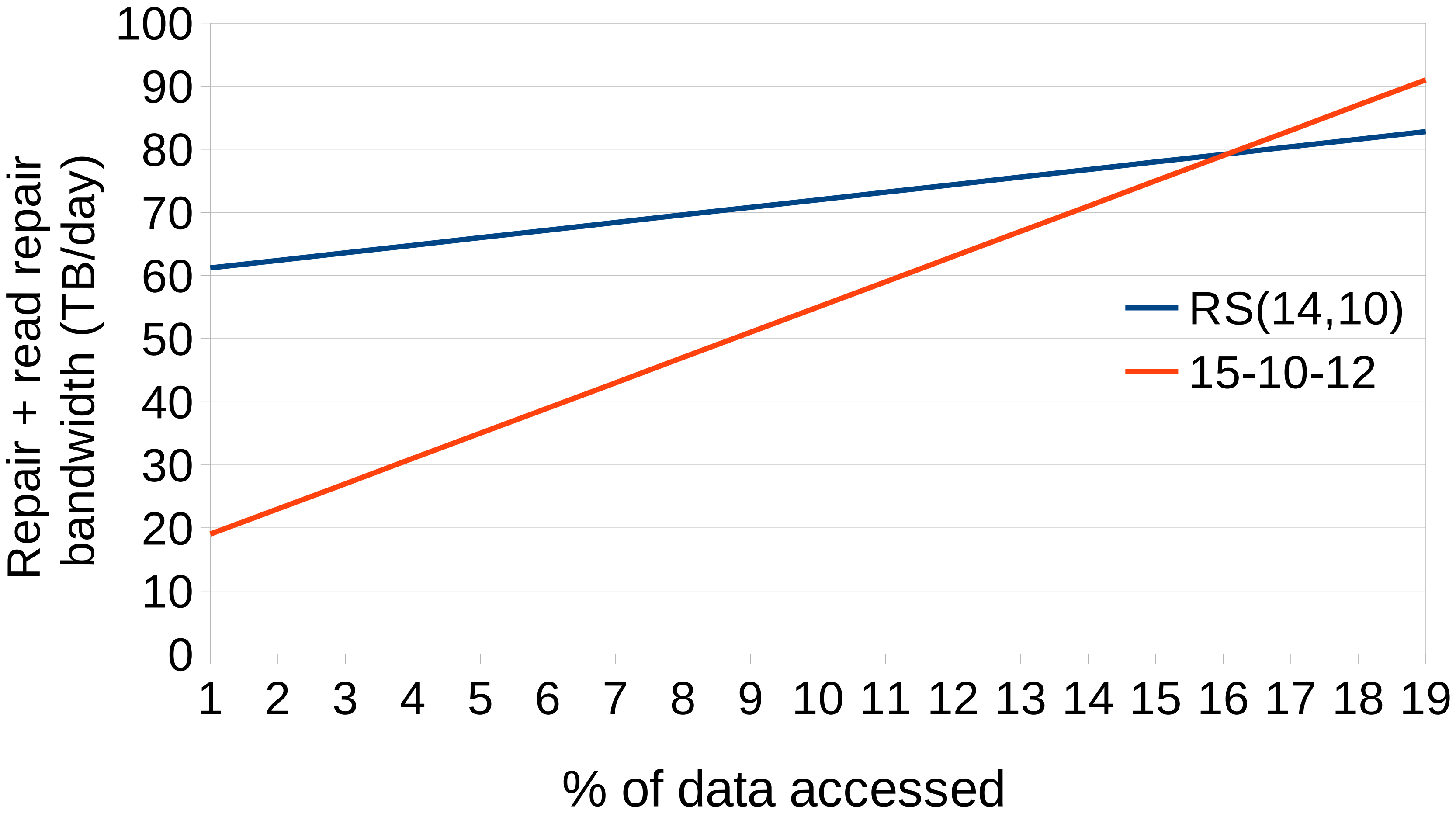
Lazy recovery is more efficient than replication



Lazy recovery is 300 times more reliable than 3-way replication



Lazy recovery wins if less than 15% data gets accessed



Summary

- Lazy recovery makes RS attractive for cold storage:
  - twice less storage, 300x better reliability, 30% lower bandwidth vs. replication
- Lazy recovery is complementary to repair-efficient coding schemes
- DS-Sim enables long-term analysis of coding schemes in large-scale systems

Summary

- Lazy recovery makes RS attractive for cold storage:
  - twice less storage, 300x better reliability, 30% lower bandwidth vs. replication
- Lazy recovery is complementary to repair-efficient coding schemes
- DS-Sim enables long-term analysis of coding schemes in large-scale systems

Thank you!