



Asclepius

Duplication and Heterogeneity-Aware Data Repair in Erasure-Coded Storage Systems

PRESENTER

Mohammad Rizk

Inria · Univ. Rennes · CNRS · IRISA

Shadi Ibrahim

Inria · Univ. Rennes · CNRS · IRISA

Thomas Lambert

Univ. Lorraine · CNRS · Inria · LORIA

This work is part of the **Inria-Hive Alvearium** collaboration.

Talk overview

01 Context

Erasure coding, Reed–Solomon, why repairing is costly, and duplication.

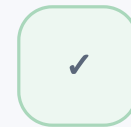
02 Asclepius

Duplication-aware repair cost, bandwidth heterogeneity, and Max–Min scheduling.

03 Evaluation

Duplication · Heterogeneity · Performance · Overhead.

Same fault-tolerance goal, lower storage overhead



Fault tolerance

Keep data available despite node failures.

x3

Replication

full copies



Higher storage overhead

OR

N+K

Erasure coding

coded redundancy



Lower storage overhead

Same objective Fault tolerance → less redundancy Erasure coding

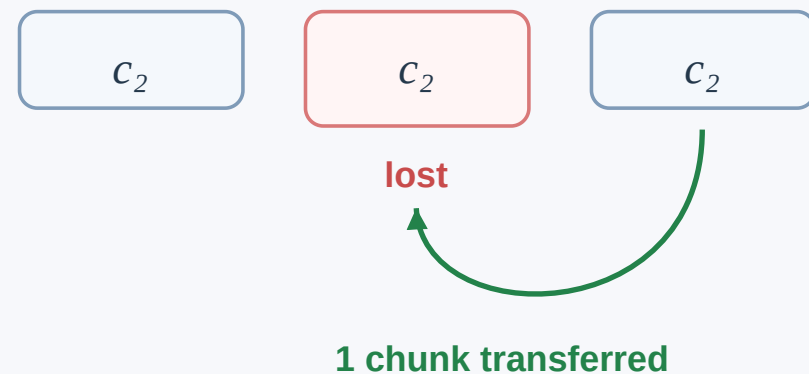
WIDELY USED



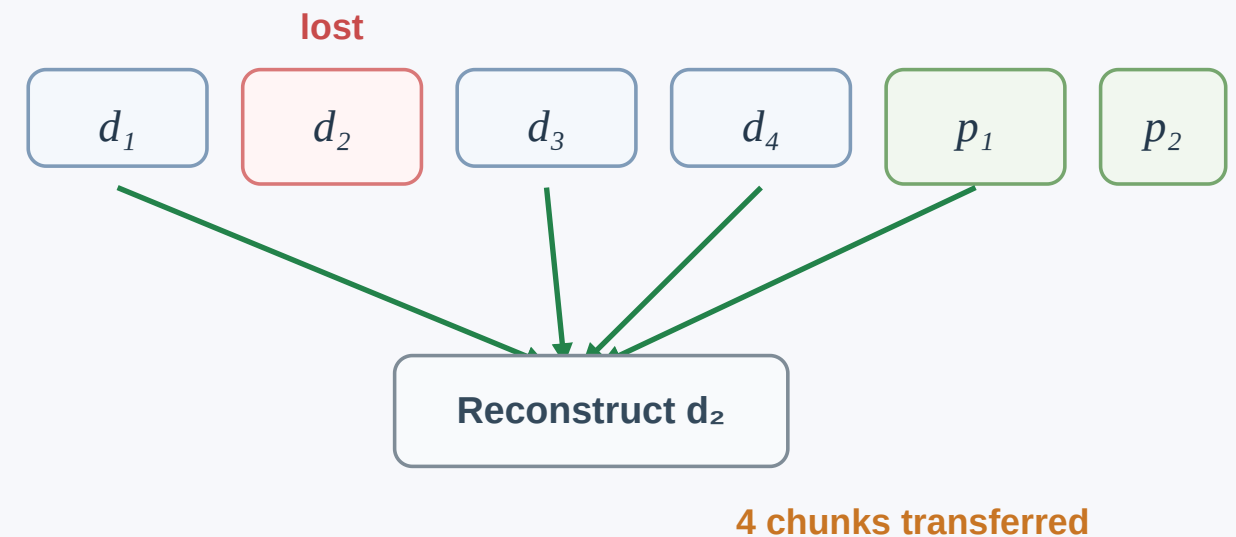
Reed–Solomon erasure coding

Reed–Solomon $RS(N,K)$ encodes N data chunks into $N+K$ chunks by adding K parity chunks. Any N surviving chunks are sufficient to reconstruct the original data.

Replication



RS(4,2) example



Compared with replication, EC provides high reliability with lower storage overhead, but repairing is costly.

Repairing one lost chunk normally requires retrieving N surviving chunks .

Shard repair: where should repair tasks run?

Erasure-coded storage systems such as **HDFS** and **IPFS (ALTOCUMULUS)** organize data into **blocks/shards**, where each block/shard contains multiple **chunks**.

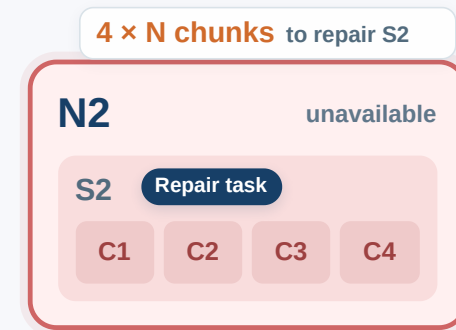


[1] Z. Shen et al., "A Survey of the Past, Present, and Future of Erasure Coding for Storage Systems," *ACM Trans. Storage*, 21(1), 2025.

[2] A. El-Shimi et al., "Primary Data Deduplication—Large Scale Study and System Design," *USENIX ATC '12*, 2012.

Shard repair: where should repair tasks run?

Erasure-coded storage systems such as **HDFS** and **IPFS (ALTOCUMULUS)** organize data into **blocks/shards**, where each block/shard contains multiple **chunks**.



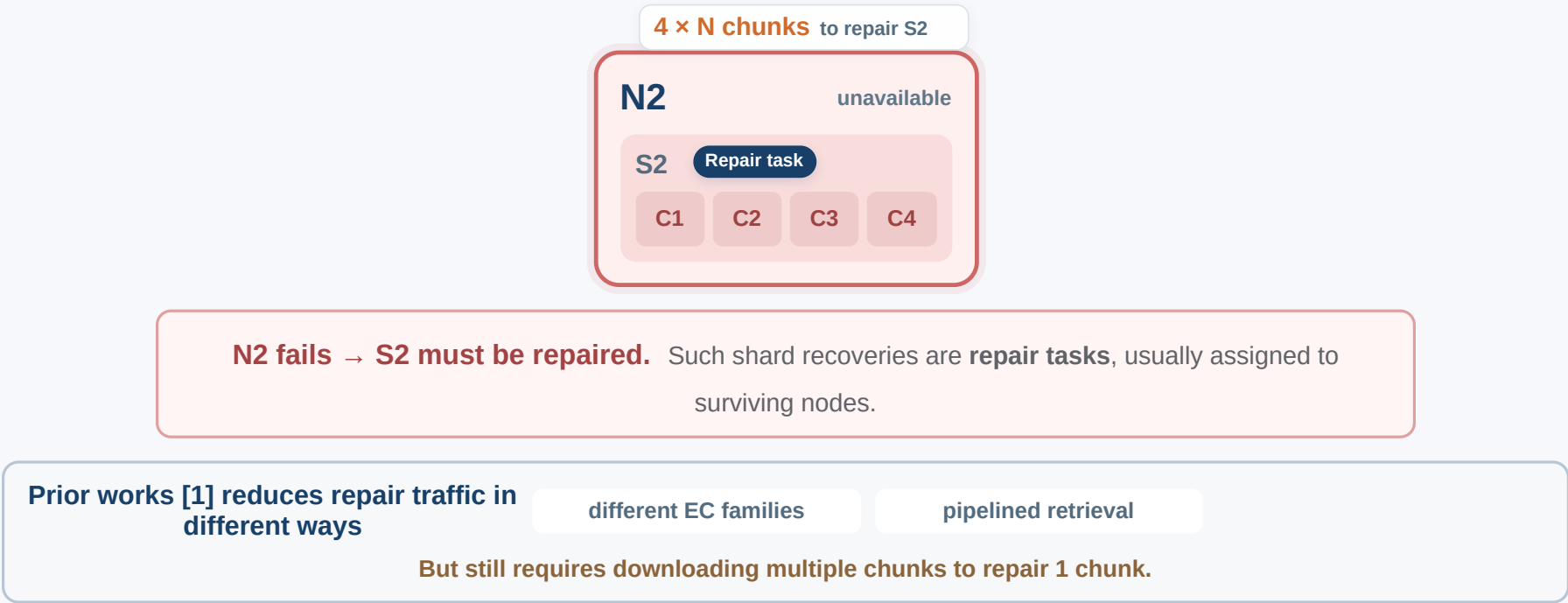
N2 fails → S2 must be repaired. Such shard recoveries are **repair tasks**, usually assigned to surviving nodes.

[1] Z. Shen et al., "A Survey of the Past, Present, and Future of Erasure Coding for Storage Systems," *ACM Trans. Storage*, 21(1), 2025.

[2] A. El-Shimi et al., "Primary Data Deduplication—Large Scale Study and System Design," *USENIX ATC '12*, 2012.

Shard repair: where should repair tasks run?

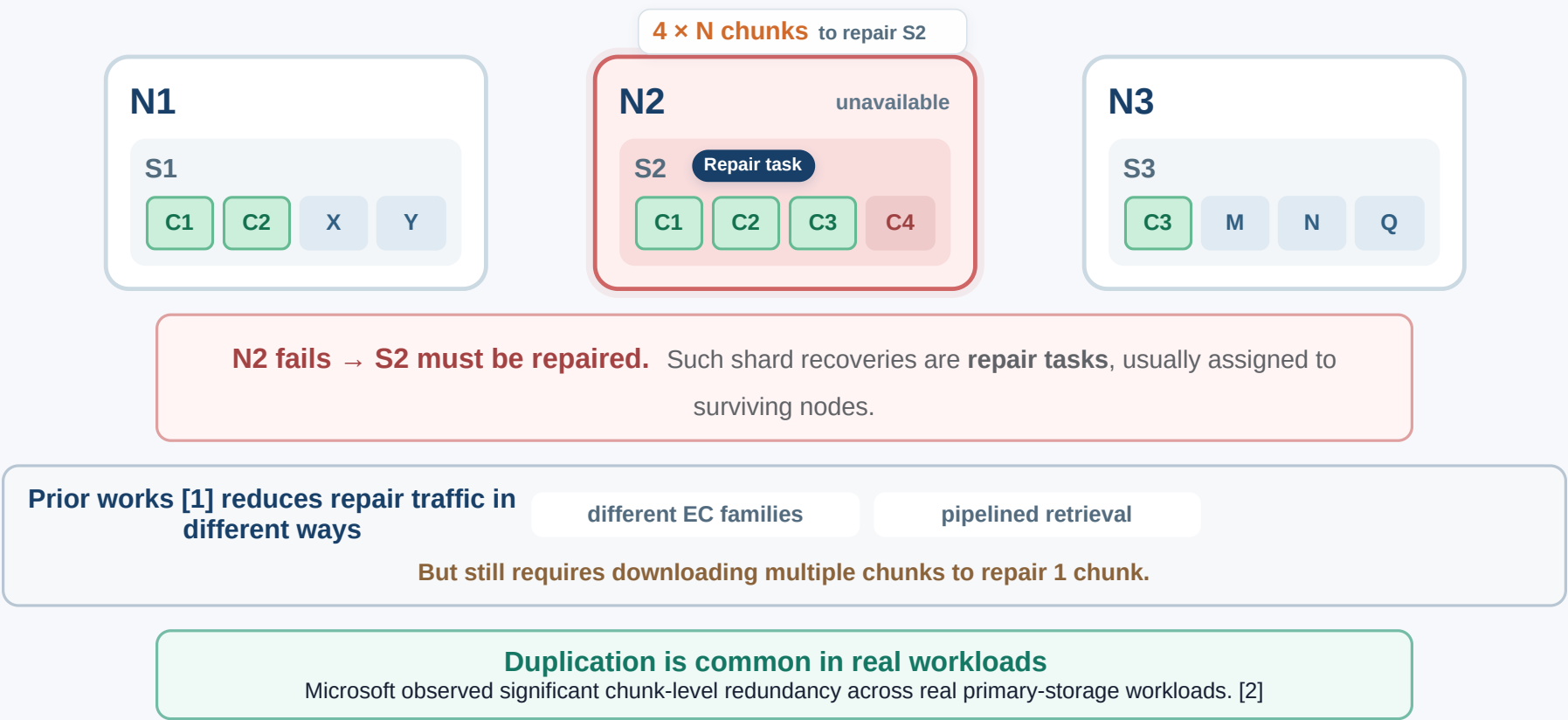
Erasure-coded storage systems such as **HDFS** and **IPFS (ALTOCUMULUS)** organize data into **blocks/shards**, where each block/shard contains multiple **chunks**.



[1]Z. Shen et al., ``A Survey of the Past, Present, and Future of Erasure Coding for Storage Systems," ACM Trans. Storage, 21(1), 2025.
[2] A. El-Shimi et al., "Primary Data Deduplication—Large Scale Study and System Design," USENIX ATC '12, 2012.

Shard repair: where should repair tasks run?

Erasure-coded storage systems such as **HDFS** and **IPFS (ALTOCUMULUS)** organize data into **blocks/shards**, where each block/shard contains multiple **chunks**.

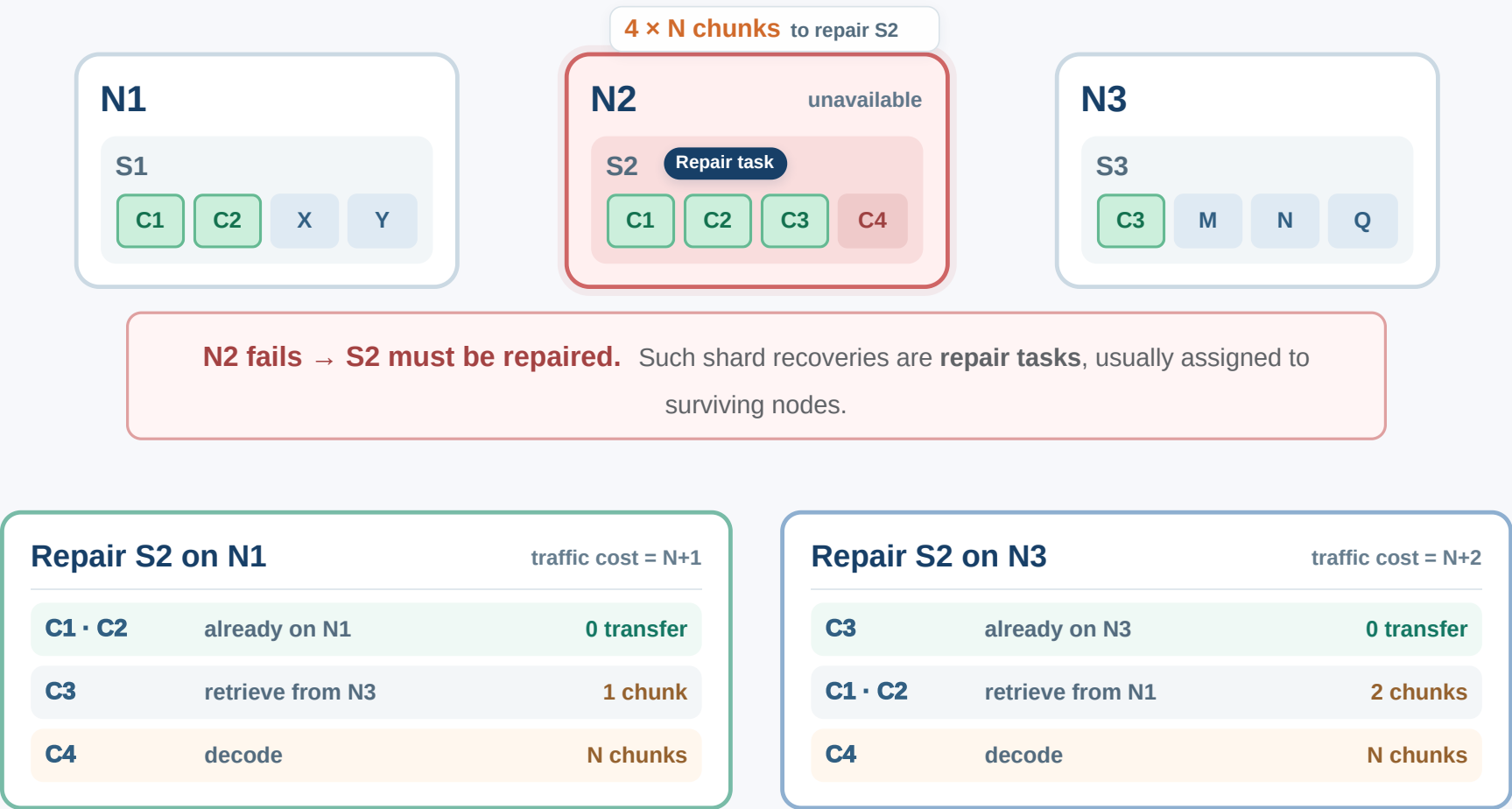


[1]Z. Shen et al., “A Survey of the Past, Present, and Future of Erasure Coding for Storage Systems,” *ACM Trans. Storage*, 21(1), 2025.

[2] A. El-Shimi et al., “Primary Data Deduplication—Large Scale Study and System Design,” *USENIX ATC '12*, 2012.

Shard repair: where should repair tasks run?

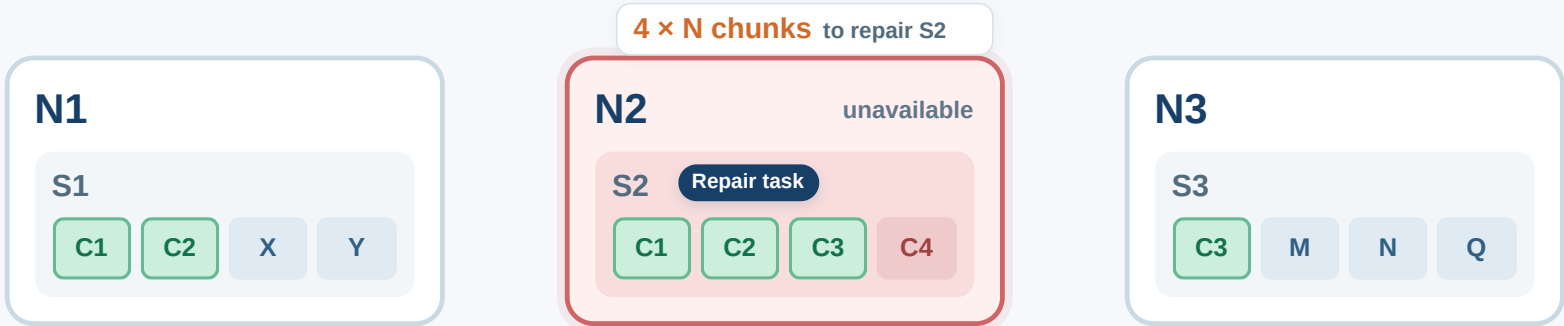
Erasure-coded storage systems such as **HDFS** and **IPFS (ALTOCUMULUS)** organize data into **blocks/shards**, where each block/shard contains multiple **chunks**.



[1]Z. Shen et al., “A Survey of the Past, Present, and Future of Erasure Coding for Storage Systems,” *ACM Trans. Storage*, 21(1), 2025.
[2] A. El-Shimi et al., “Primary Data Deduplication—Large Scale Study and System Design,” *USENIX ATC '12*, 2012.

Shard repair: where should repair tasks run?

Erasure-coded storage systems such as **HDFS** and **IPFS (ALTOCUMULUS)** organize data into **blocks/shards**, where each block/shard contains multiple **chunks**.



N2 fails → S2 must be repaired. Such shard recoveries are **repair tasks**, usually assigned to surviving nodes.

Repair S2 on N1			traffic cost = $N+1$
C1 · C2	already on N1	0 transfer	
C3	retrieve from N3	1 chunk	
C4	decode	N chunks	

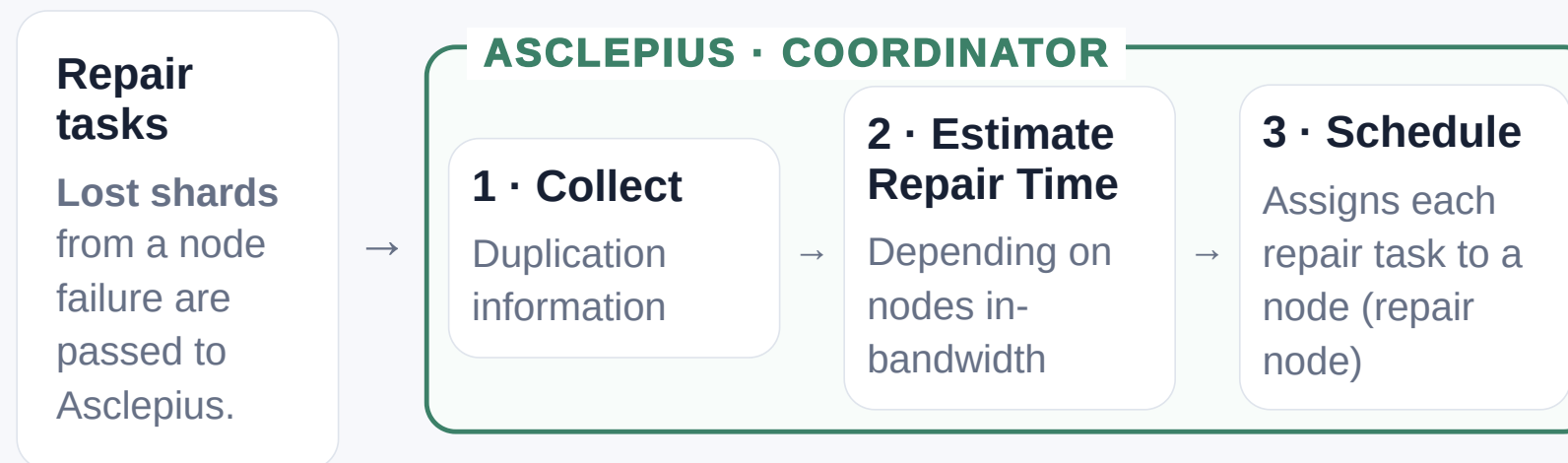
Repair S2 on N3			traffic cost = $N+2$
C3	already on N3	0 transfer	
C1 · C2	retrieve from N1	2 chunks	
C4	decode	N chunks	

Duplication reduces repair traffic. But lower traffic alone is not enough: repair-task scheduling must also account for node heterogeneity.

[1]Z. Shen et al., “A Survey of the Past, Present, and Future of Erasure Coding for Storage Systems,” *ACM Trans. Storage*, 21(1), 2025.
[2] A. El-Shimi et al., “Primary Data Deduplication—Large Scale Study and System Design,” *USENIX ATC '12*, 2012.

Asclepius: duplication- and heterogeneity-aware repair scheduling

Asclepius is a system that uses duplicated data chunks to steer repairs in erasure-coded storage systems.



Asclepius decides where each repair task should be executed.

From global knowledge to repair-time estimation

ASCLEPIUS COORDINATOR

From global knowledge to repair-time estimation

ASCLEPIUS COORDINATOR

Lost shard S1

c1

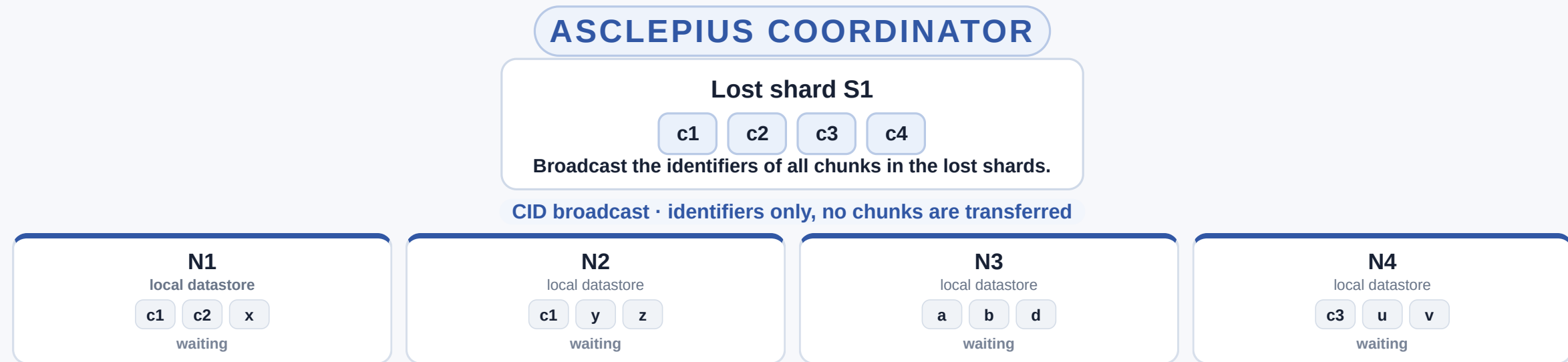
c2

c3

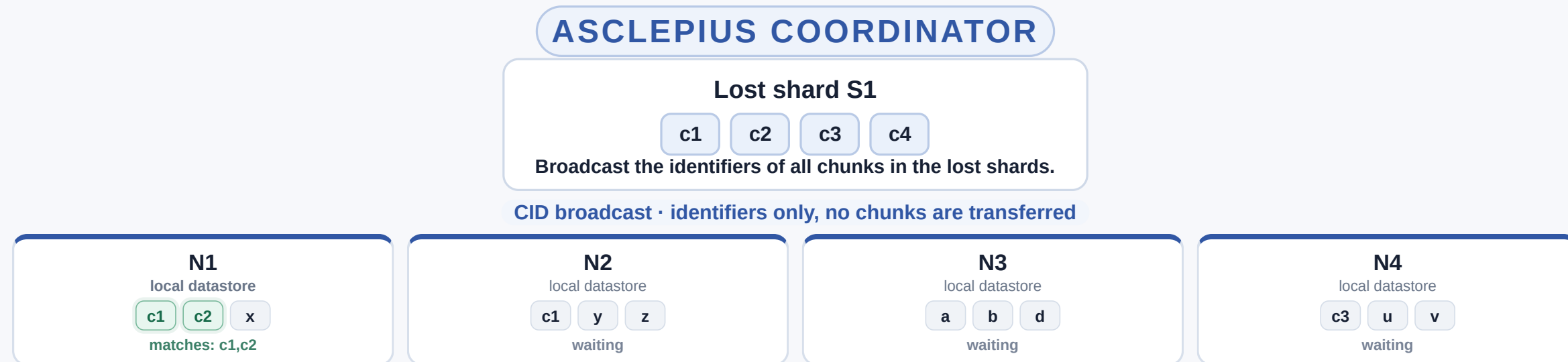
c4

Broadcast the identifiers of all chunks in the lost shards.

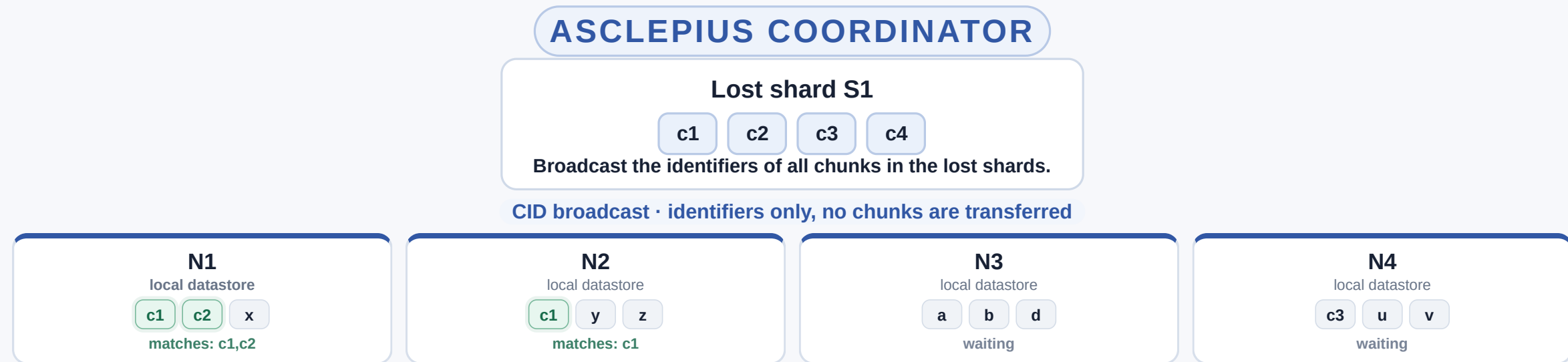
From global knowledge to repair-time estimation



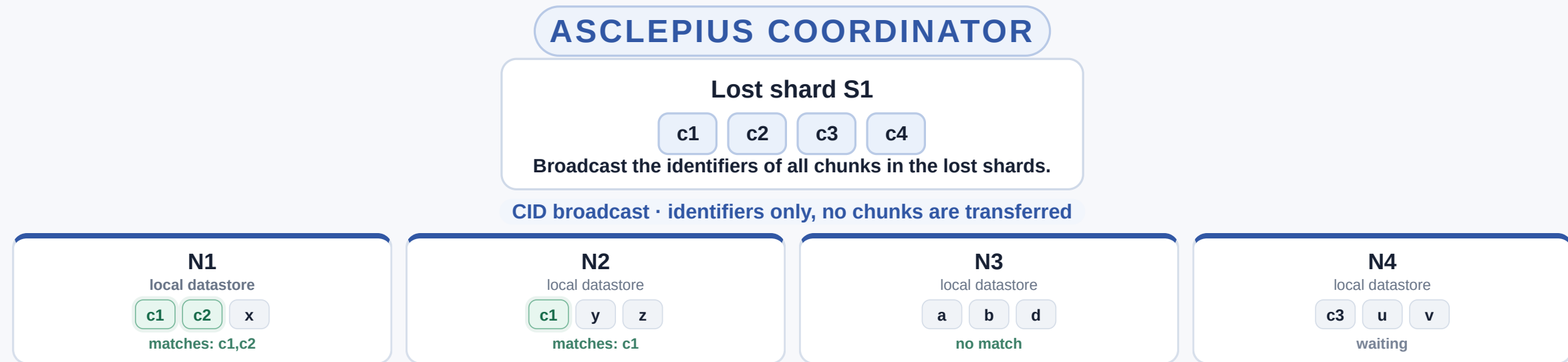
From global knowledge to repair-time estimation



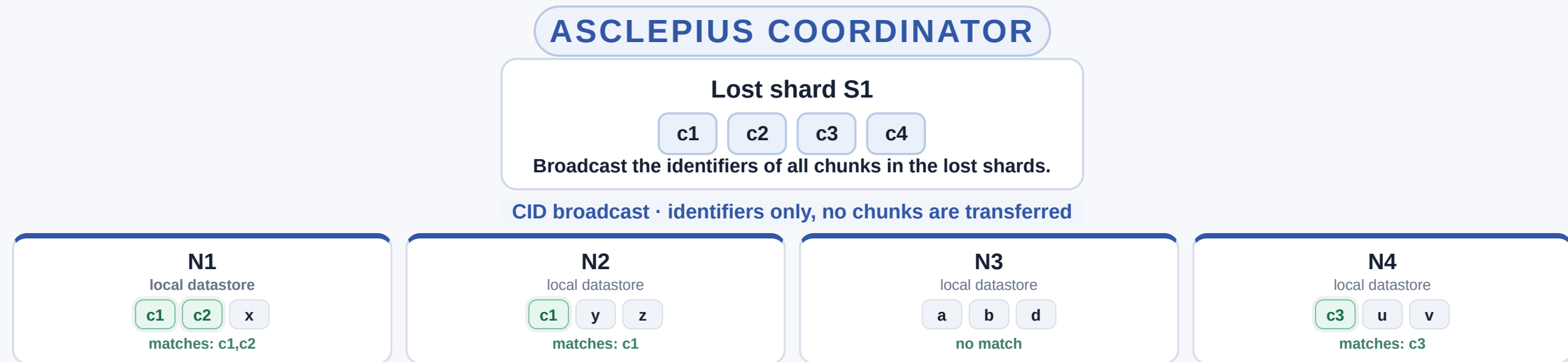
From global knowledge to repair-time estimation



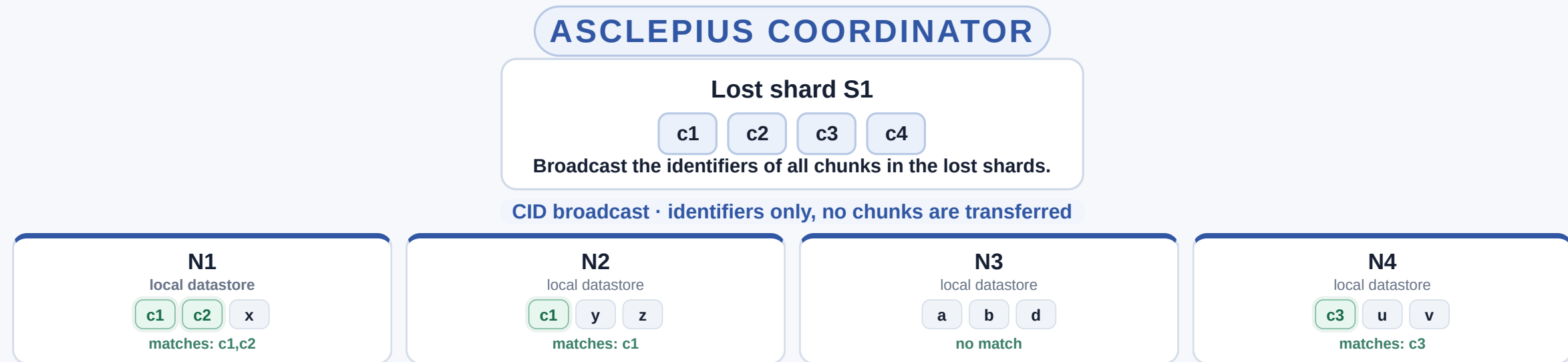
From global knowledge to repair-time estimation



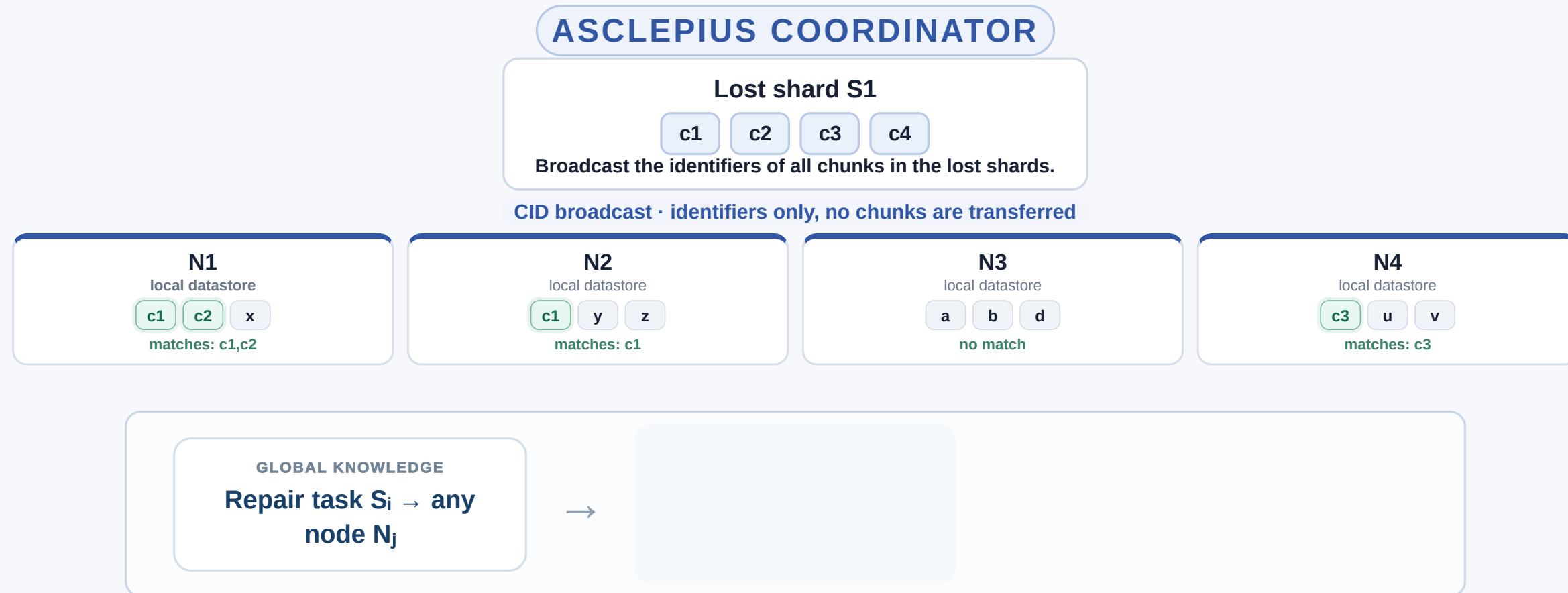
From global knowledge to repair-time estimation



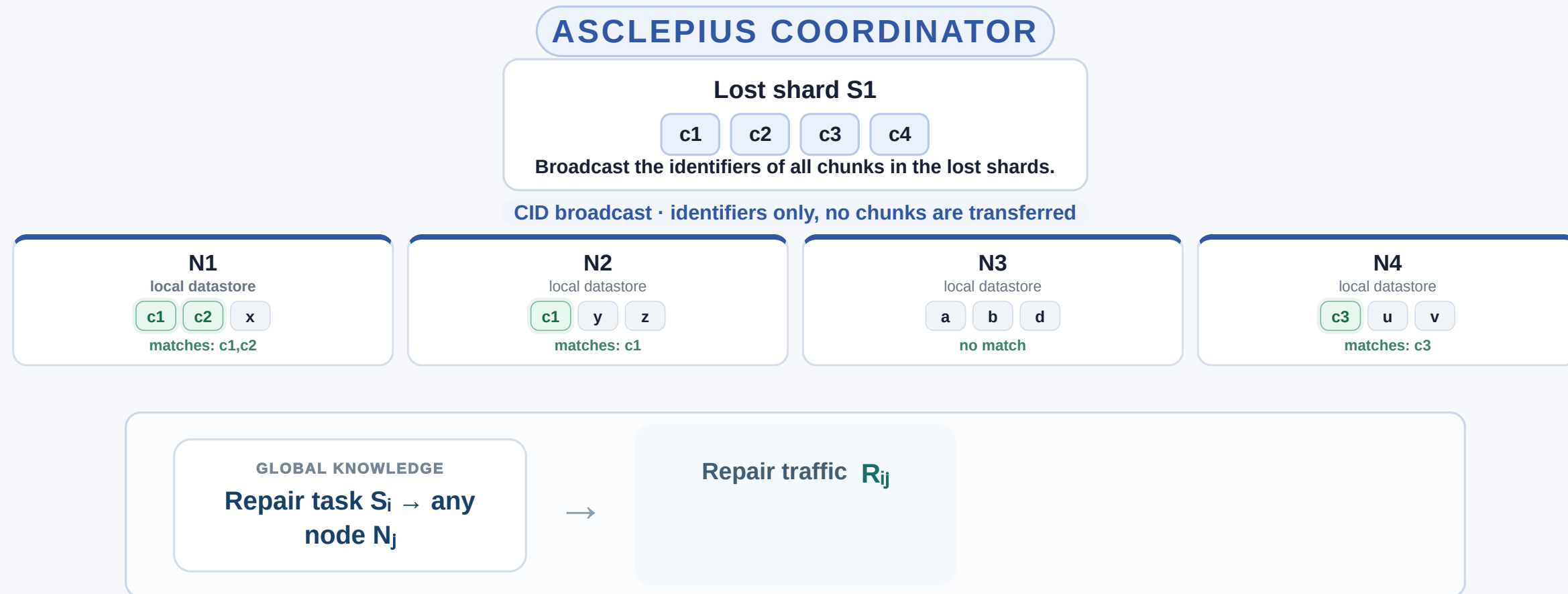
From global knowledge to repair-time estimation



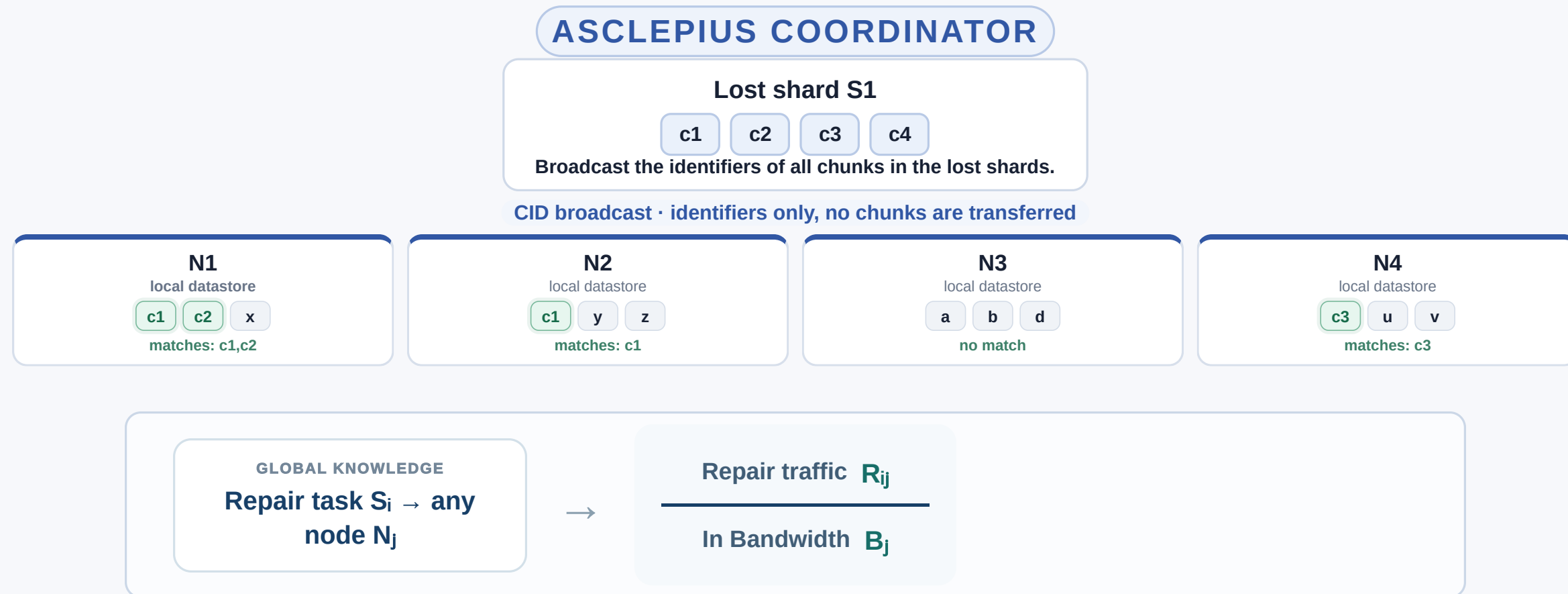
From global knowledge to repair-time estimation



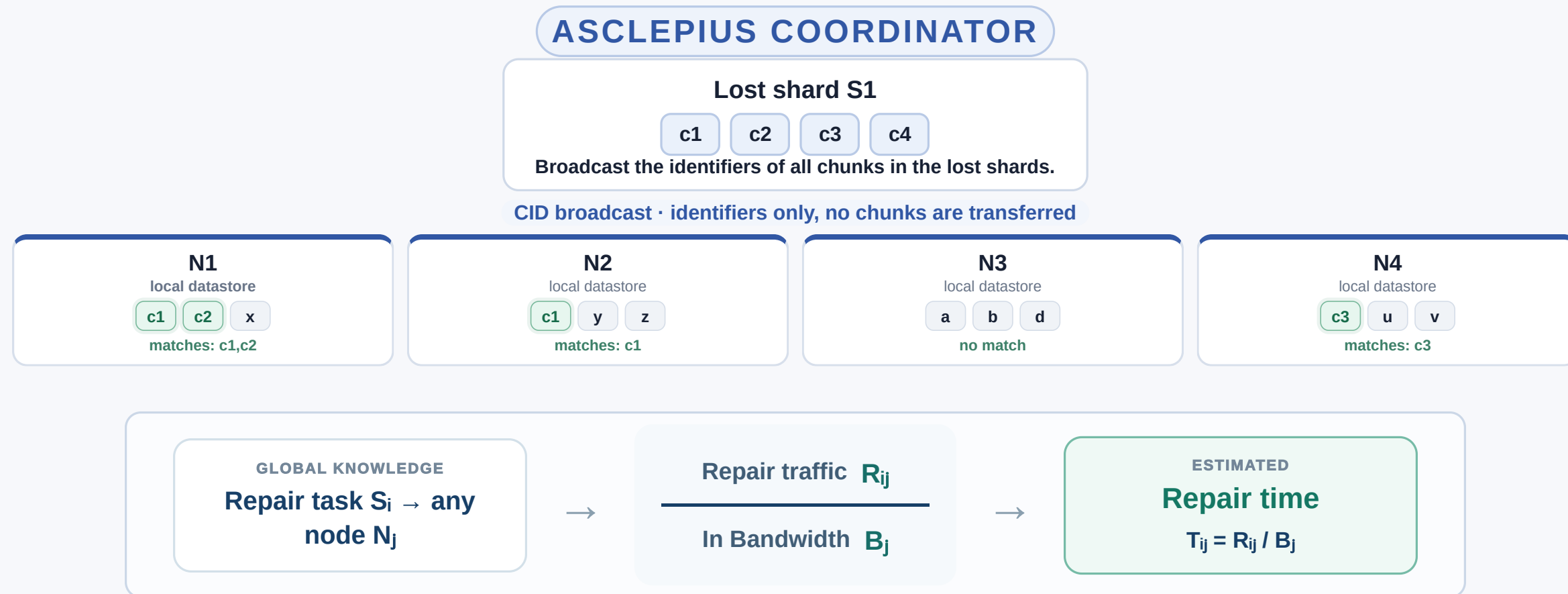
From global knowledge to repair-time estimation



From global knowledge to repair-time estimation

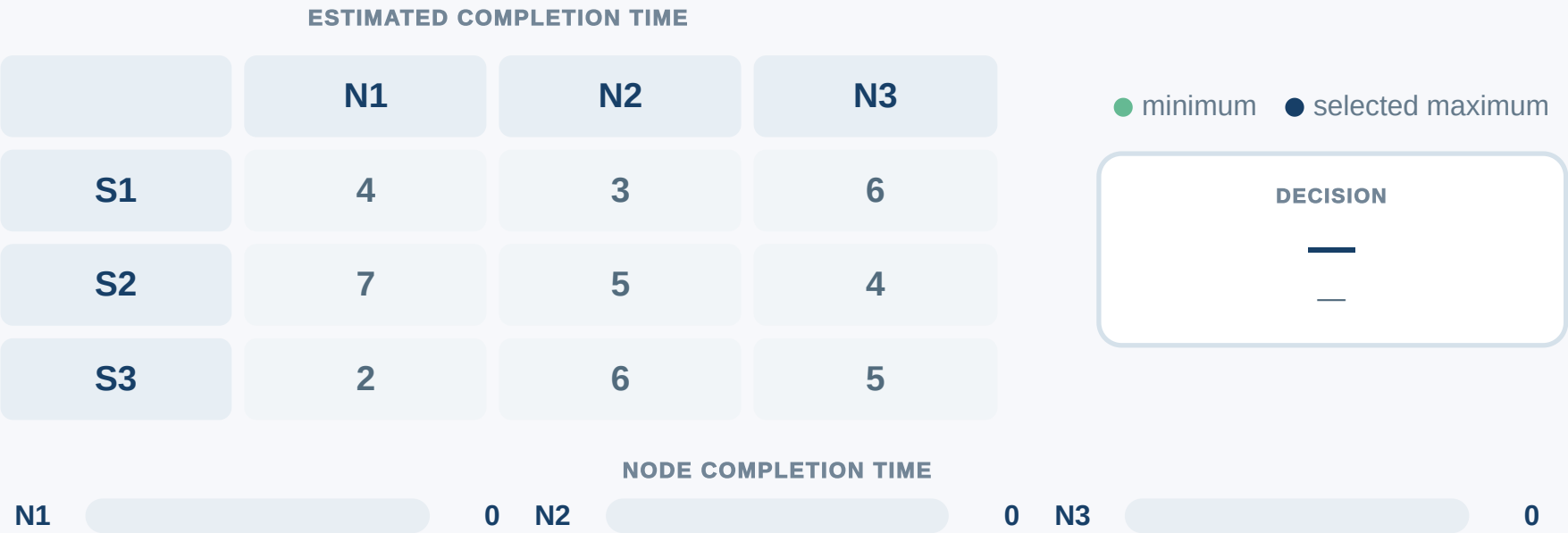


From global knowledge to repair-time estimation



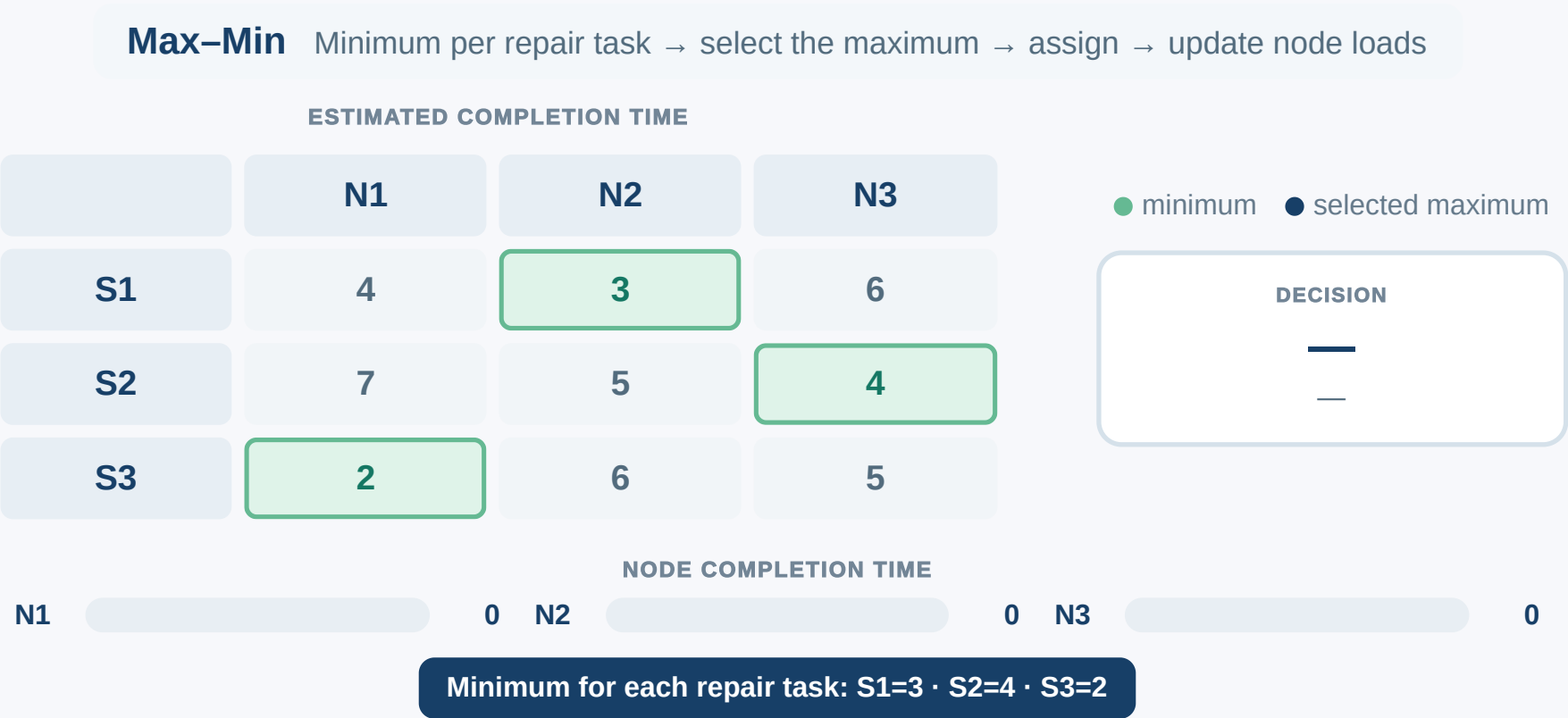
Max–Min repair scheduling

Max–Min Minimum per repair task → select the maximum → assign → update node loads

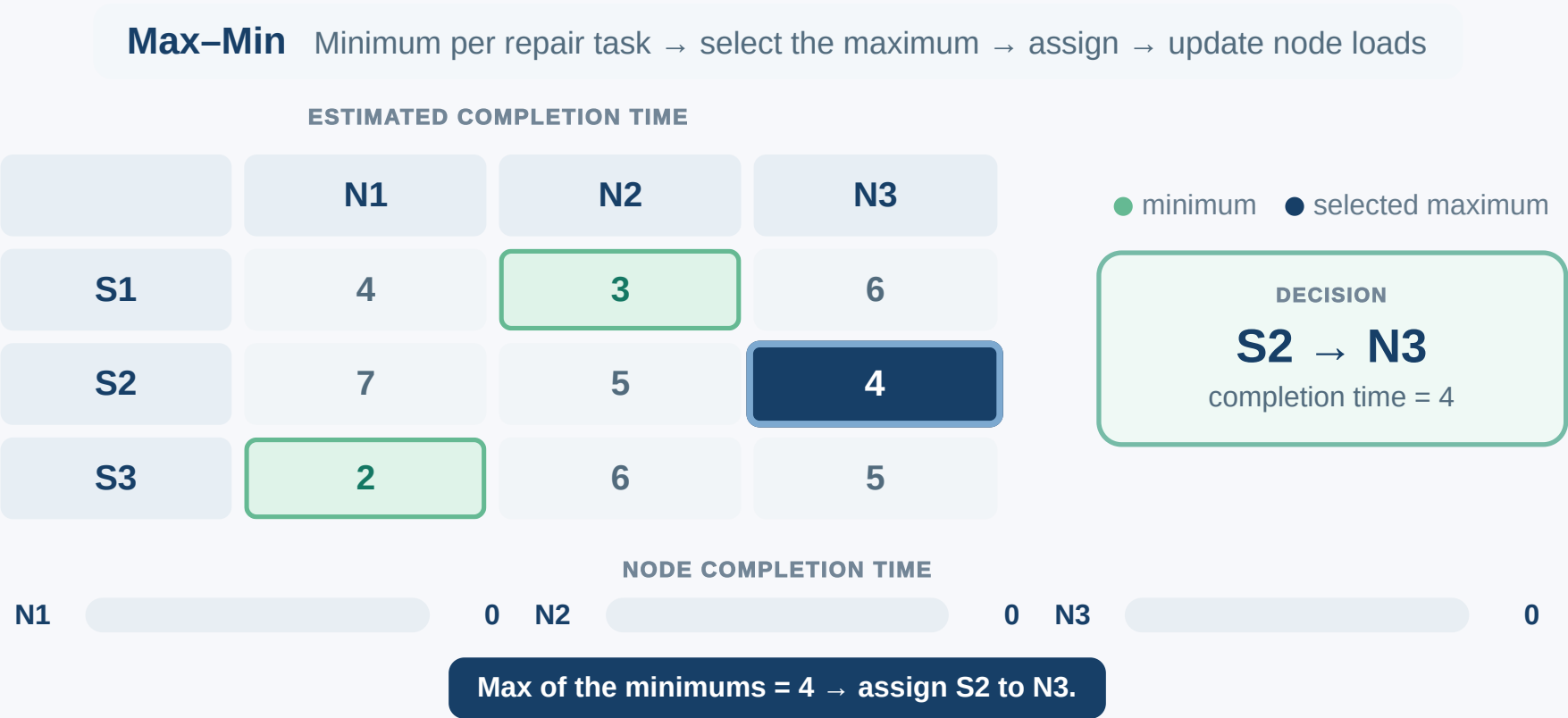


Start with the estimated completion-time matrix.

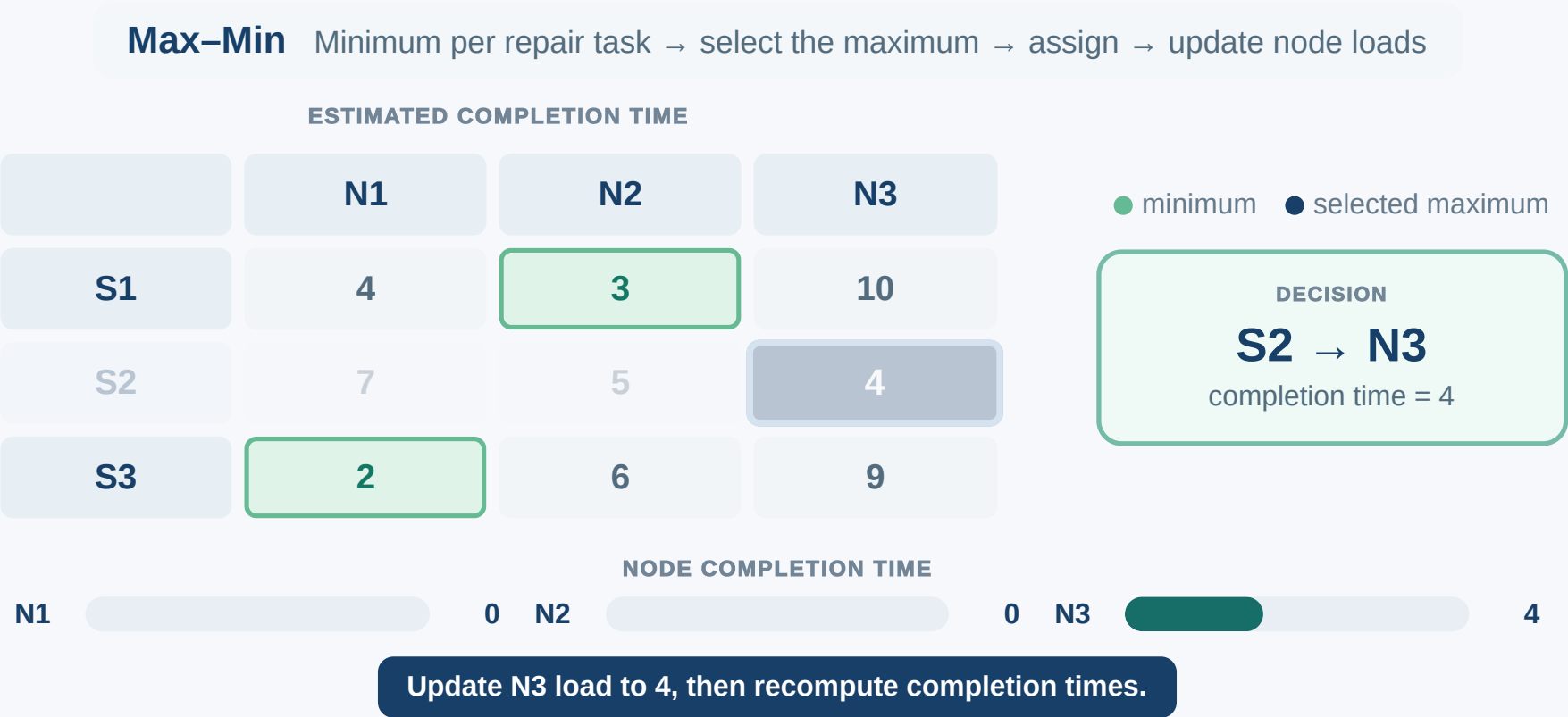
Max–Min repair scheduling



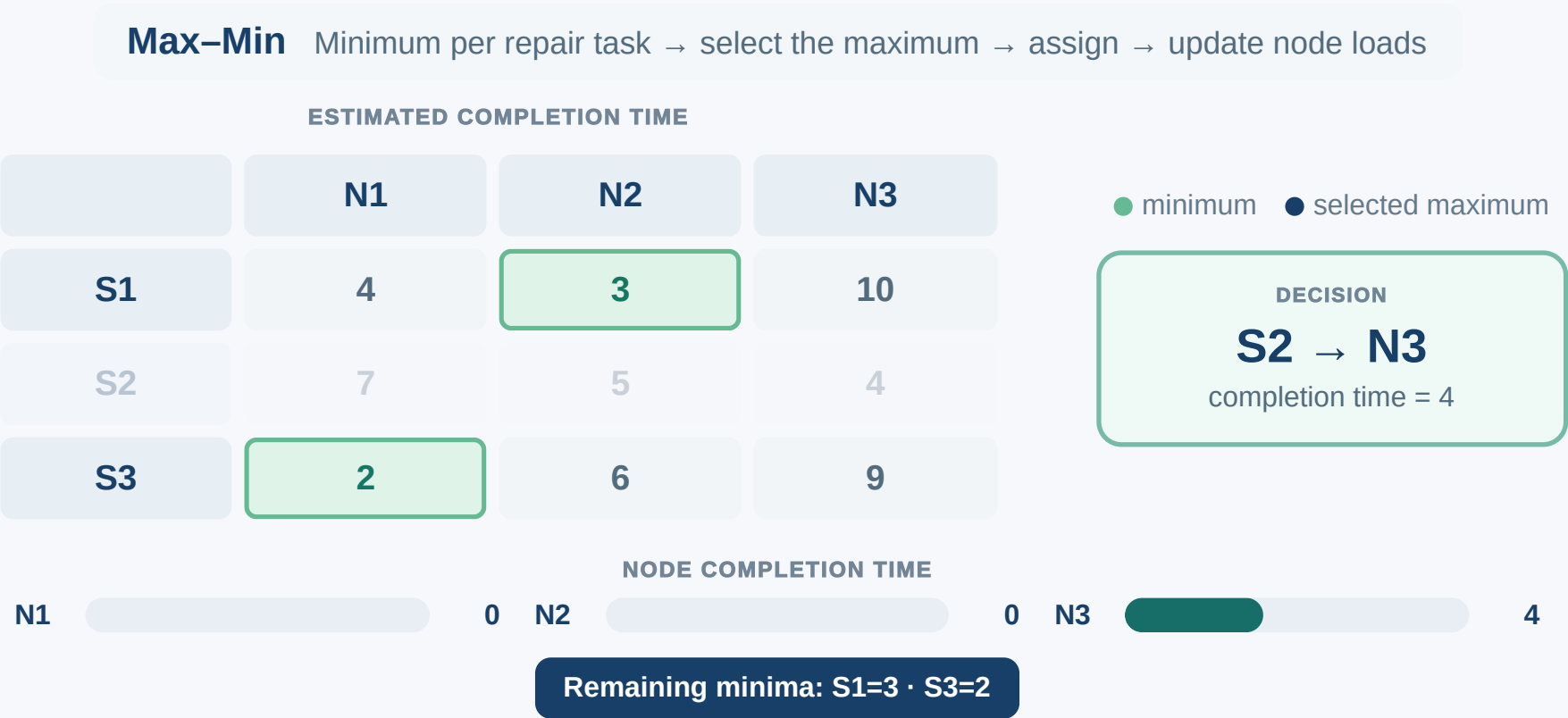
Max–Min repair scheduling



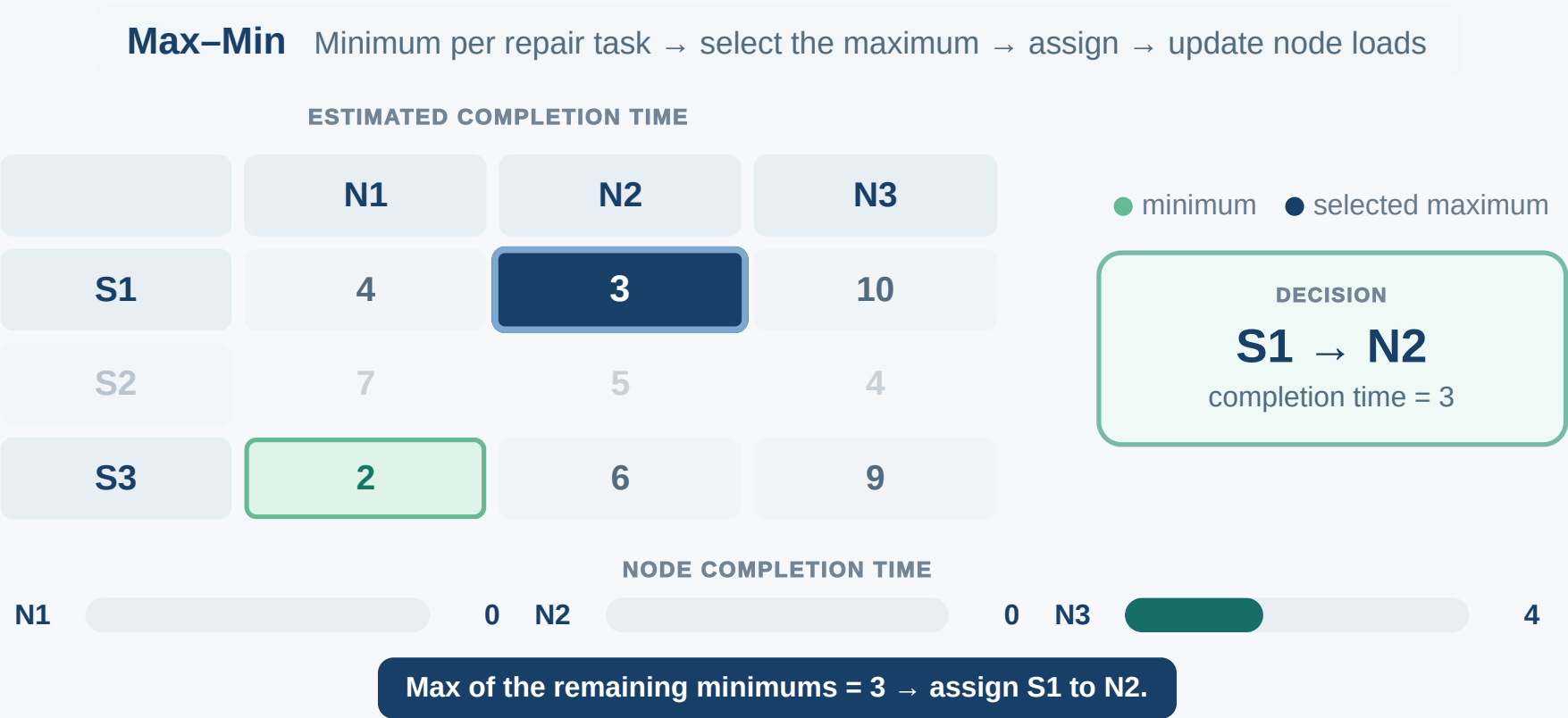
Max–Min repair scheduling



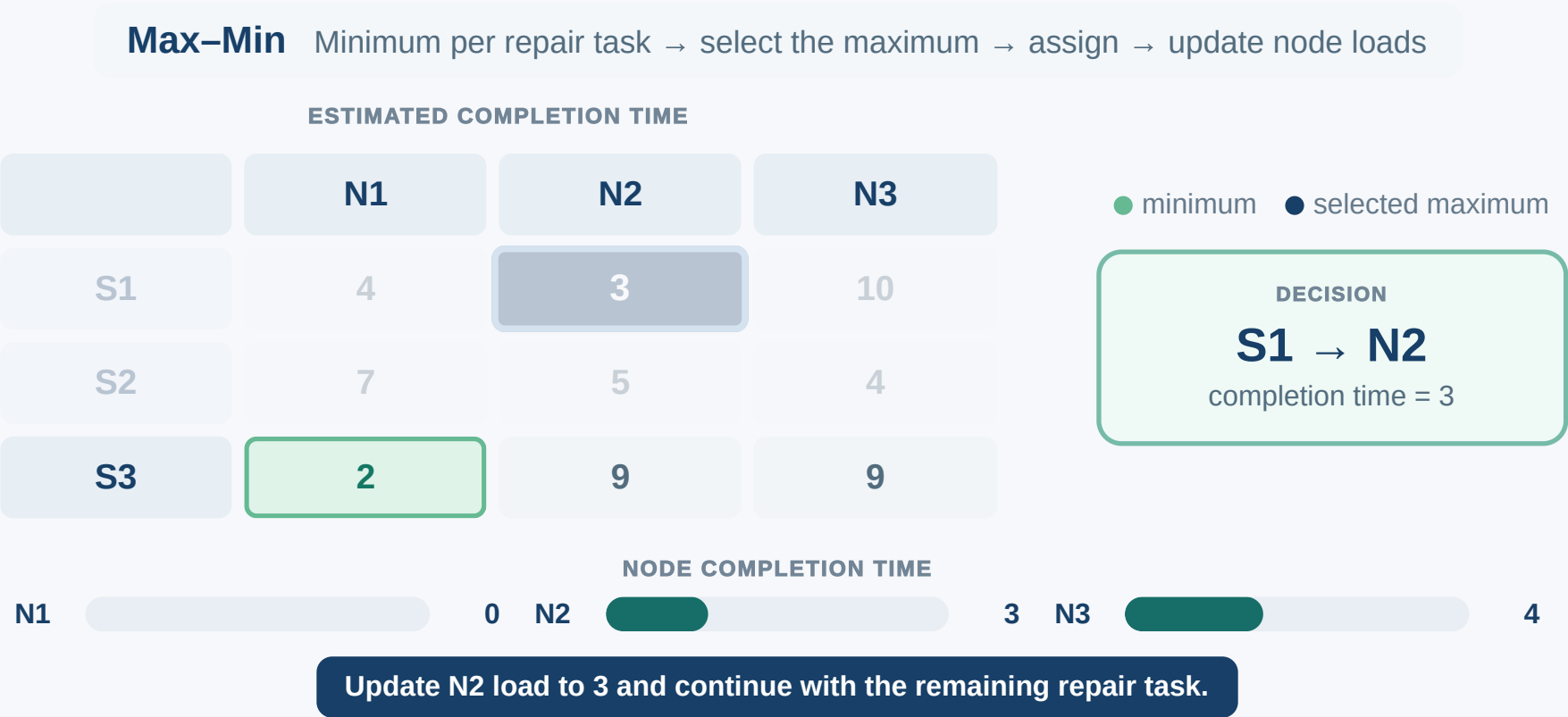
Max–Min repair scheduling



Max–Min repair scheduling



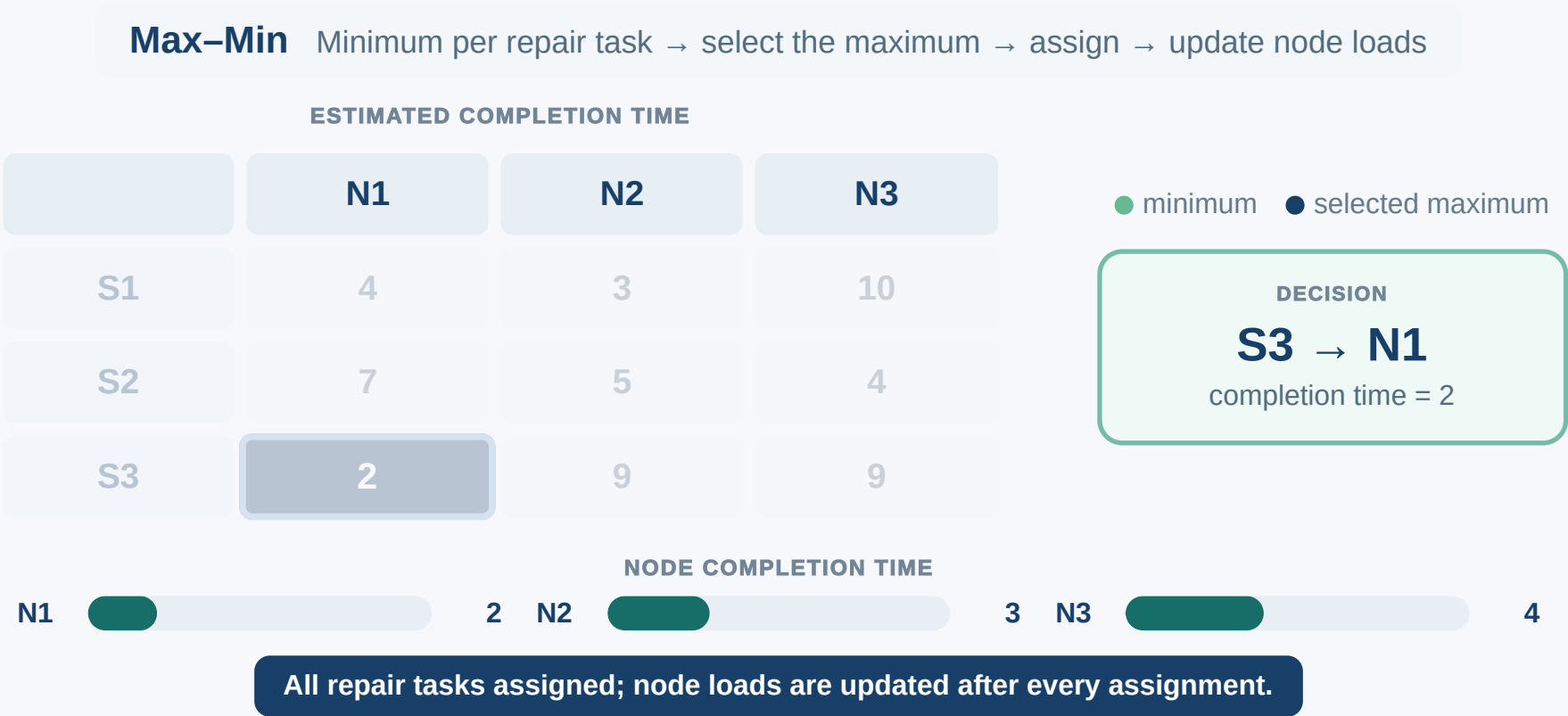
Max–Min repair scheduling



Max–Min repair scheduling



Max–Min repair scheduling



Prototype evaluation on SLICES-FR

Implementation & evaluation Asclepius is implemented and evaluated on top of **ALTOCUMULUS**[3] — a P2P storage system built on IPFS with Reed–Solomon erasure coding.

26

Dahu machines
Grenoble

RS(10,4)

evaluation code

256 KB

IPFS chunks

4×

repetitions

Datasets

DS1–DS3	50 GB each · duplication 0 / 0.3 / 0.7 · 100 MB shards (Synthetic DataSets)
DS4	50 GB · duplication 0.7 · variable 20–100 MB shards (Synthetic DataSet)
DockerHub	50 GB · duplication 0.76 · variable 2–100 MB shards (RealWorld DataSet)

Network

Homogeneous	2 Gbit/s download bandwidth for all nodes
Heterogeneous	80–1384 Mbit/s download bandwidth · Internet Topology Zoo

XOR

Assign each repair task using the XOR distance between the candidate node ID and the lost-shard CID ·

BAL

Distribute repair tasks across candidate nodes in round-robin fashion · Altocumulus default

DEDUP

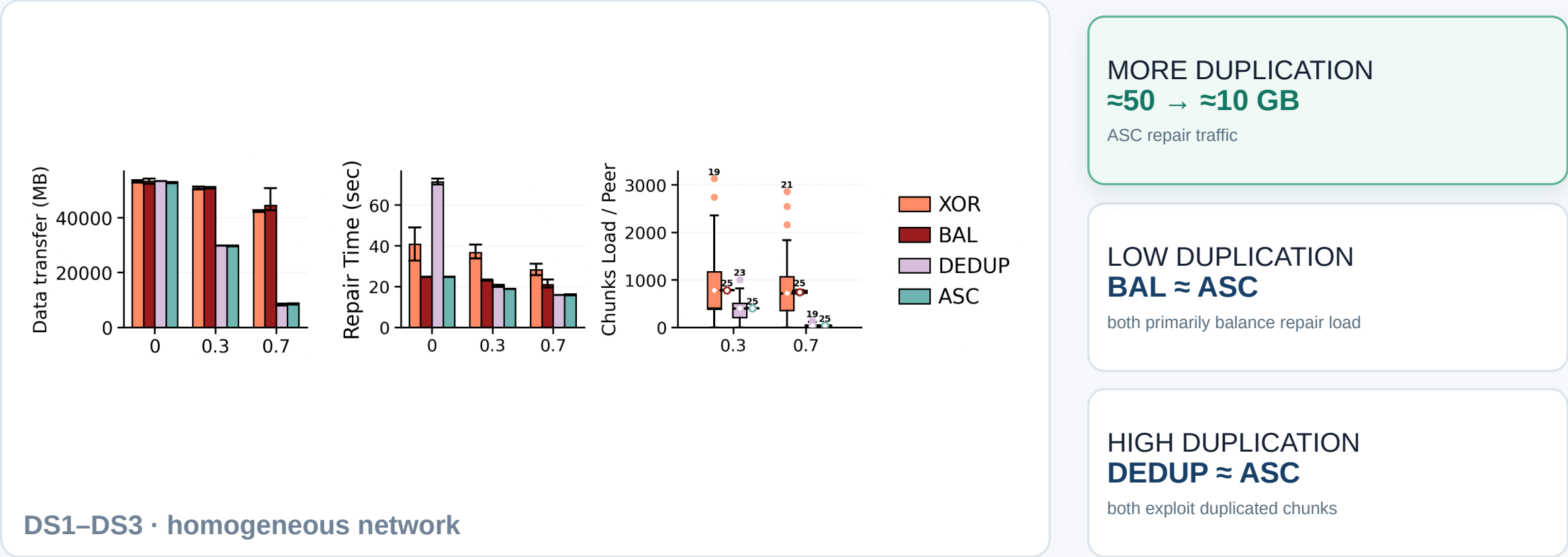
Greedy assign each repair task to the node with the highest similarity to the missing shard

ASC

Asclepius.

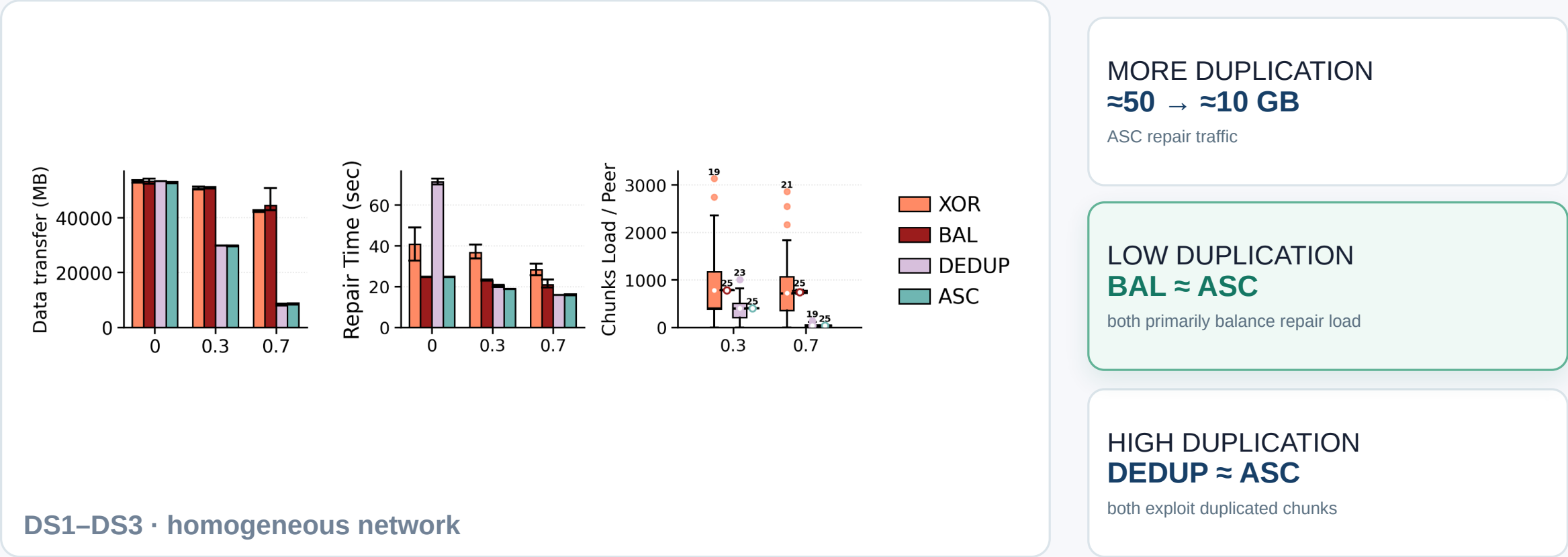
[3] M. Rizk, S. Ibrahim, and T. Lambert, ``ALTOCUMULUS: Enabling Efficient Erasure Coding in IPFS," IEEE/ACM CCGrid, 2026.

Asclepius turns duplication into lower repair traffic



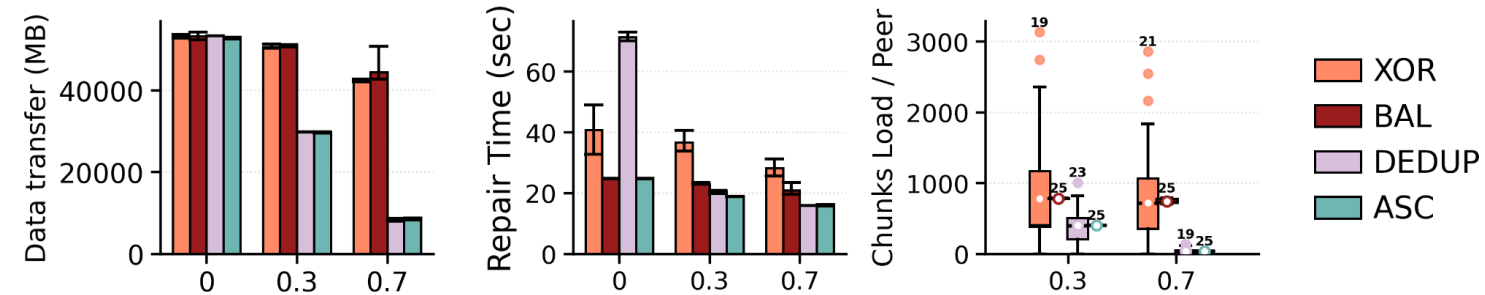
ASC leverages duplication while maintaining balanced repair loads

Asclepius turns duplication into lower repair traffic



ASC leverages duplication while maintaining balanced repair loads

Asclepius turns duplication into lower repair traffic



DS1–DS3 · homogeneous network

MORE DUPLICATION

≈50 → ≈10 GB

ASC repair traffic

LOW DUPLICATION

BAL ≈ ASC

both primarily balance repair load

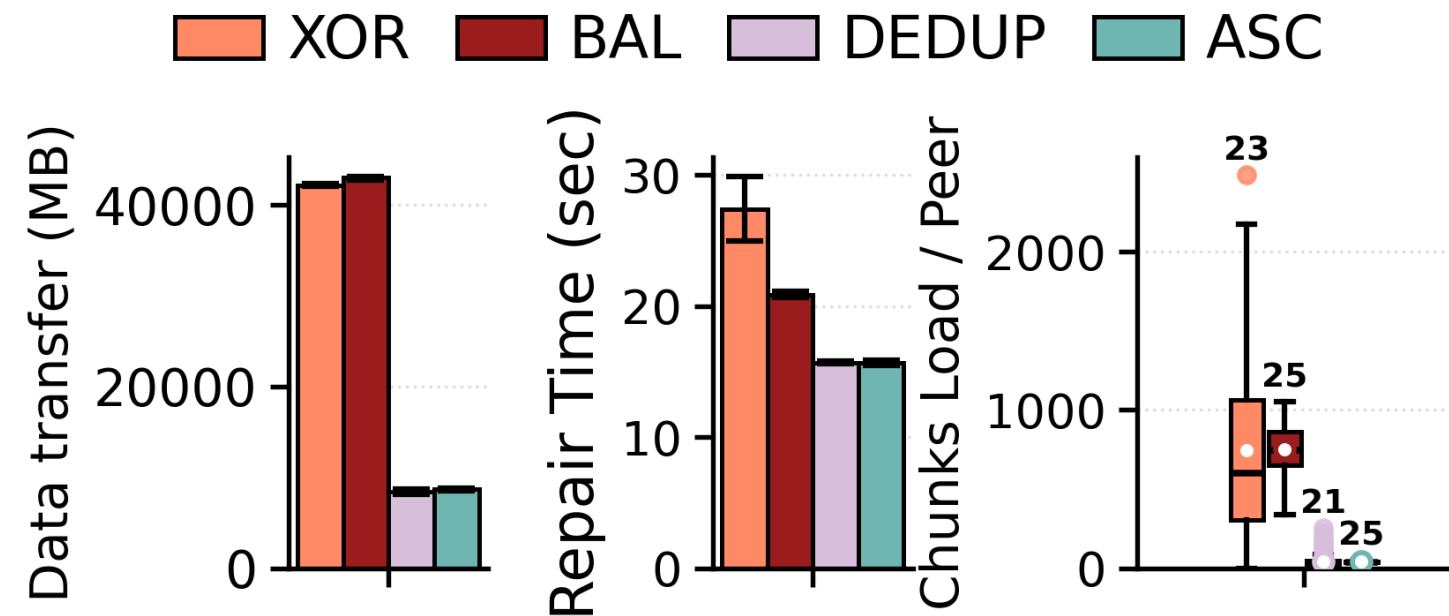
HIGH DUPLICATION

DEDUP ≈ ASC

both exploit duplicated chunks

ASC leverages duplication while maintaining balanced repair loads

Asclepius is aware of the heterogeneity in Repair-Task Sizes



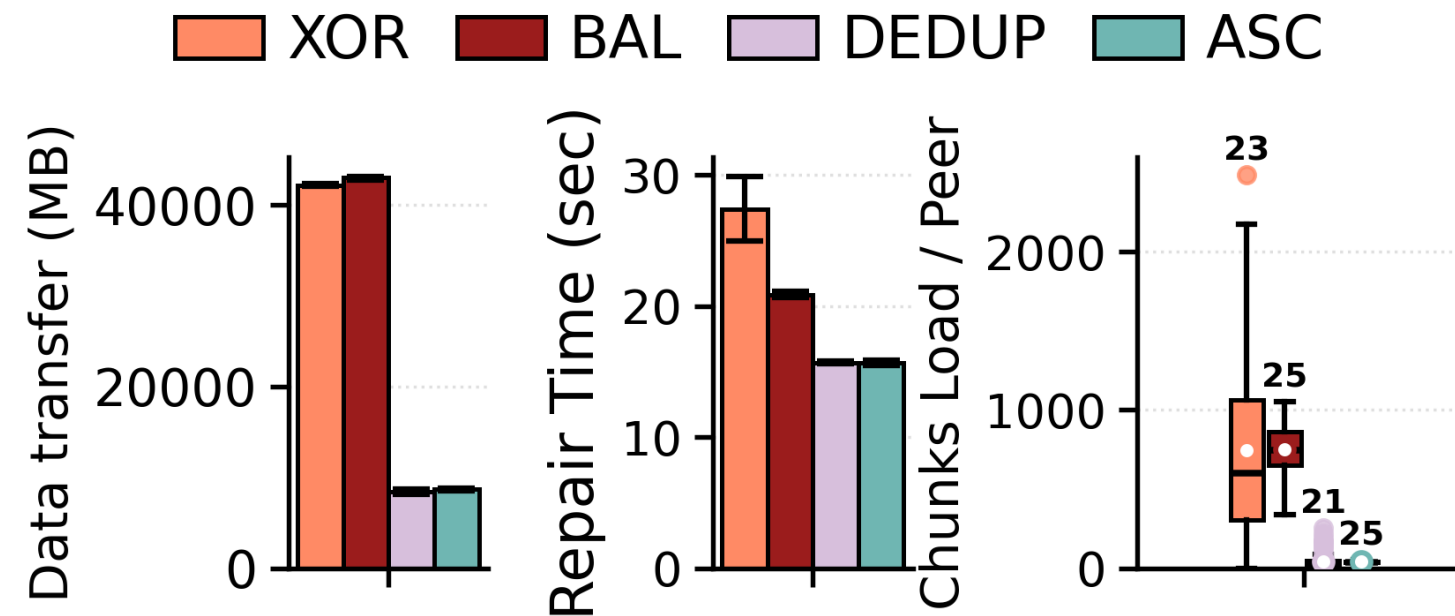
DS4 · variable shard sizes · homogeneous network

VARIABLE SHARD SIZES
up to 26% faster
ASC compared with BAL

WHY? BAL counts tasks
ASC accounts for their different
repair costs.

ASC accounts for heterogeneous repair-task sizes

Asclepius is aware of the heterogeneity in Repair-Task Sizes



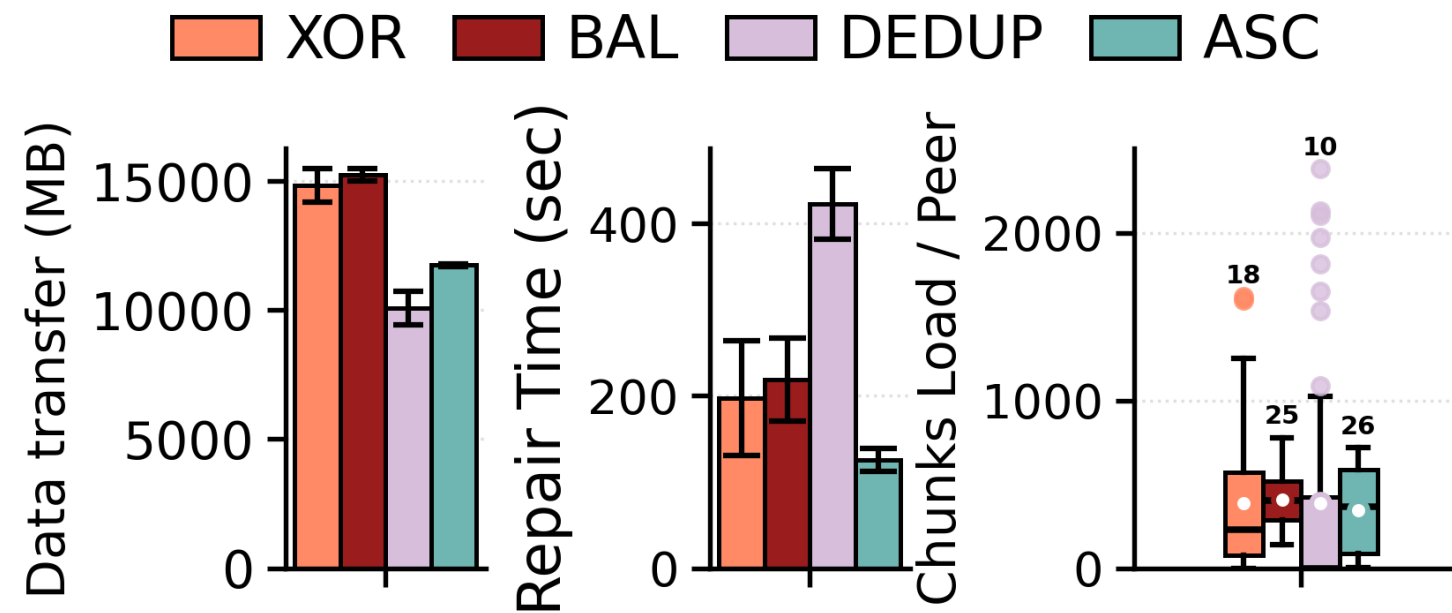
DS4 · variable shard sizes · homogeneous network

VARIABLE SHARD SIZES
up to 26% faster
ASC compared with BAL

WHY? BAL counts tasks
ASC accounts for their different
repair costs.

ASC accounts for heterogeneous repair-task sizes

Less traffic does not necessarily mean faster repair



DockerHub · heterogeneous network

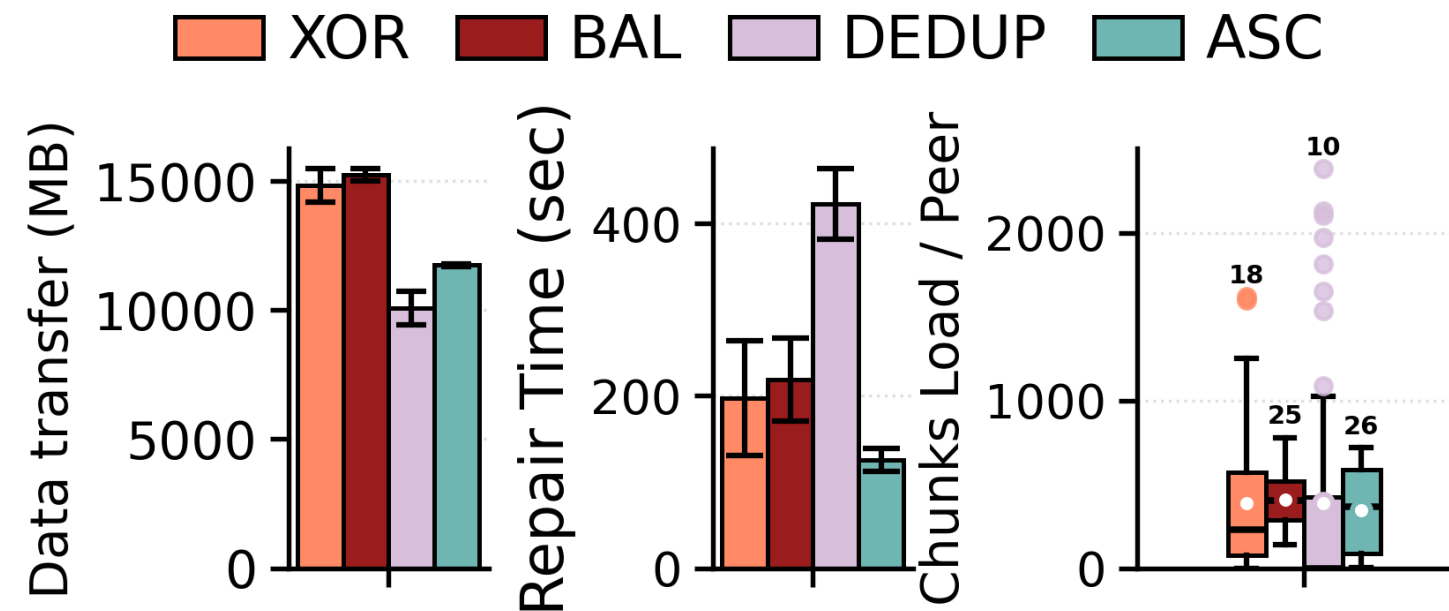
VS DEDUP
≈3× faster
ASC repair

VS BAL
≈2× faster
ASC repair

VS XOR
1.96× faster
ASC repair

ASC trades additional repair traffic for lower repair time when beneficial

Less traffic does not necessarily mean faster repair



DockerHub · heterogeneous network

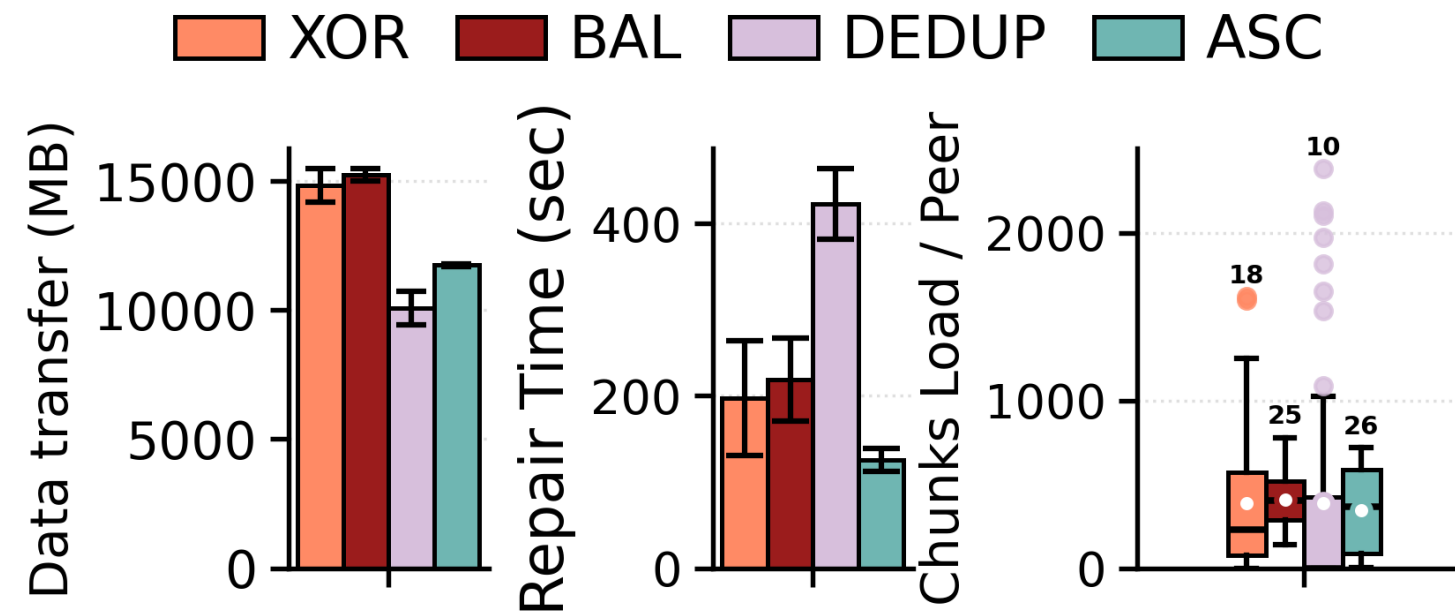
VS DEDUP
≈3× faster
ASC repair

VS BAL
≈2× faster
ASC repair

VS XOR
1.96× faster
ASC repair

ASC trades additional repair traffic for lower repair time when beneficial

Less traffic does not necessarily mean faster repair



DockerHub · heterogeneous network

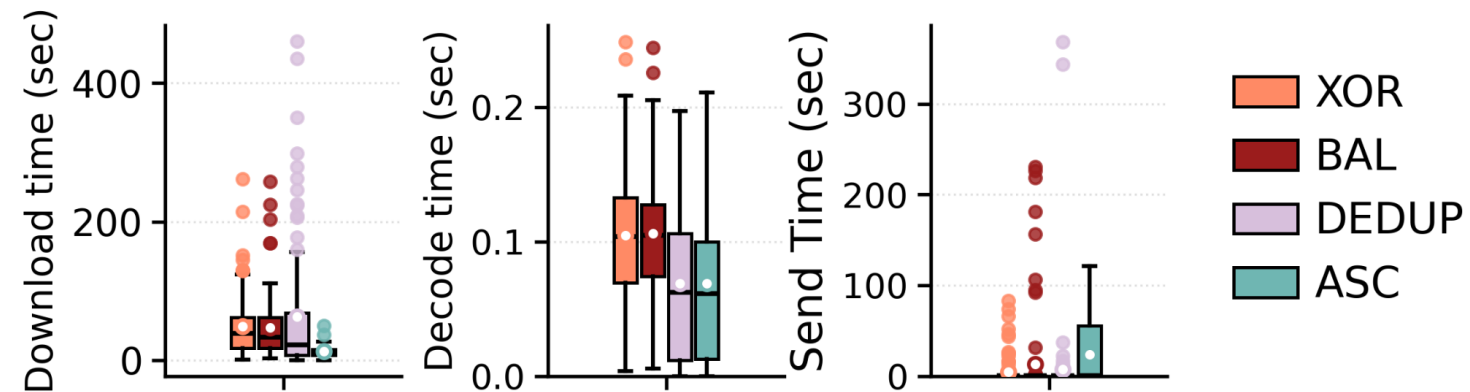
VS DEDUP
≈3× faster
ASC repair

VS BAL
≈2× faster
ASC repair

VS XOR
1.96× faster
ASC repair

ASC trades additional repair traffic for lower repair time when beneficial

Asclepius significantly reduces download time



Heterogeneous-network evaluation

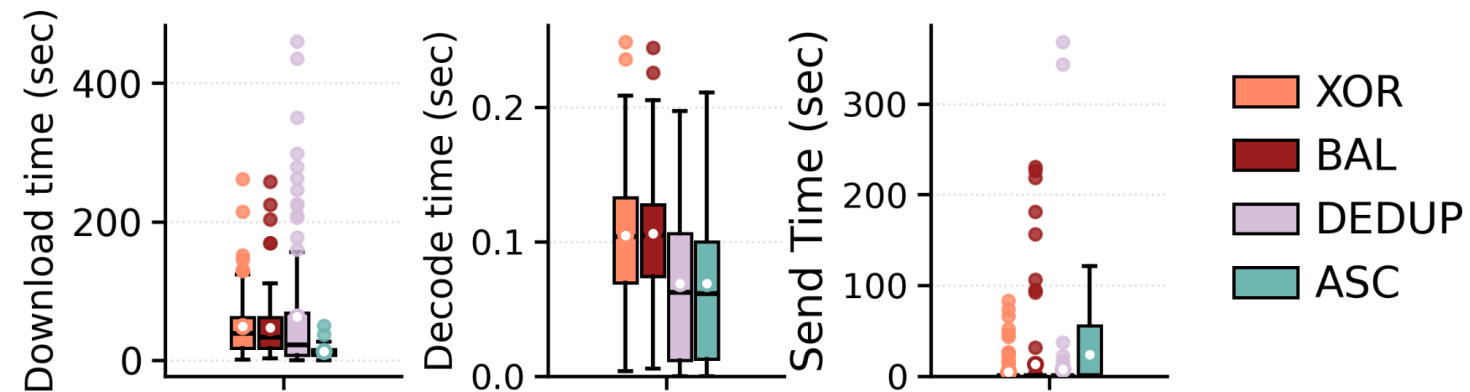
DOWNLOAD TIME
≈96% lower
than DEDUP

DOWNLOAD TIME
≈92% lower
than BAL and XOR

REMAINING LIMITATION
Send time
can dominate when the destination
node is slow.

ASC reduces download times but is not aware of placement policies in the system

Asclepius significantly reduces download time



Heterogeneous-network evaluation

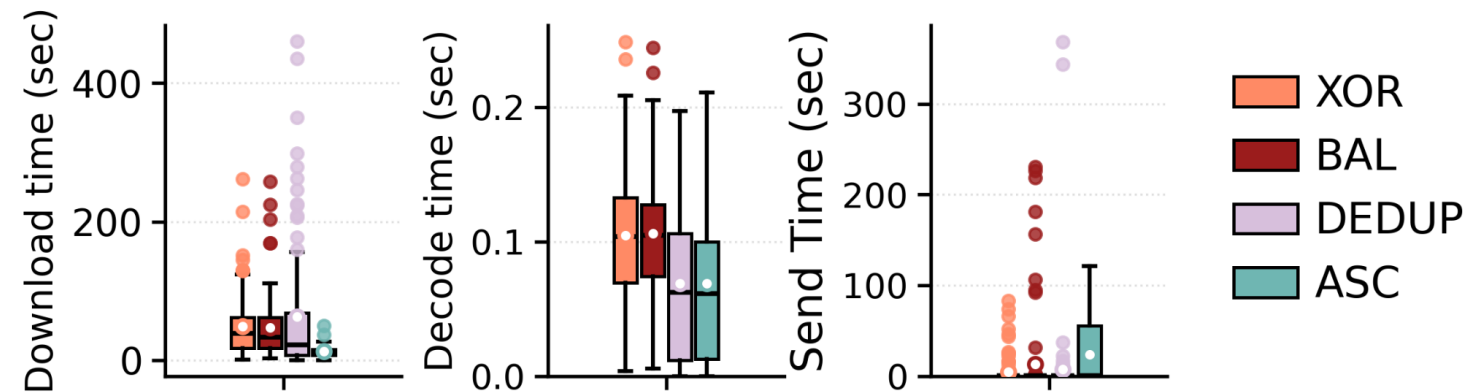
DOWNLOAD TIME
≈**96% lower**
than DEDUP

DOWNLOAD TIME
≈**92% lower**
than BAL and XOR

REMAINING LIMITATION
Send time
can dominate when the destination
node is slow.

ASC reduces download times but is not aware of placement policies in the system

Asclepius significantly reduces download time



Heterogeneous-network evaluation

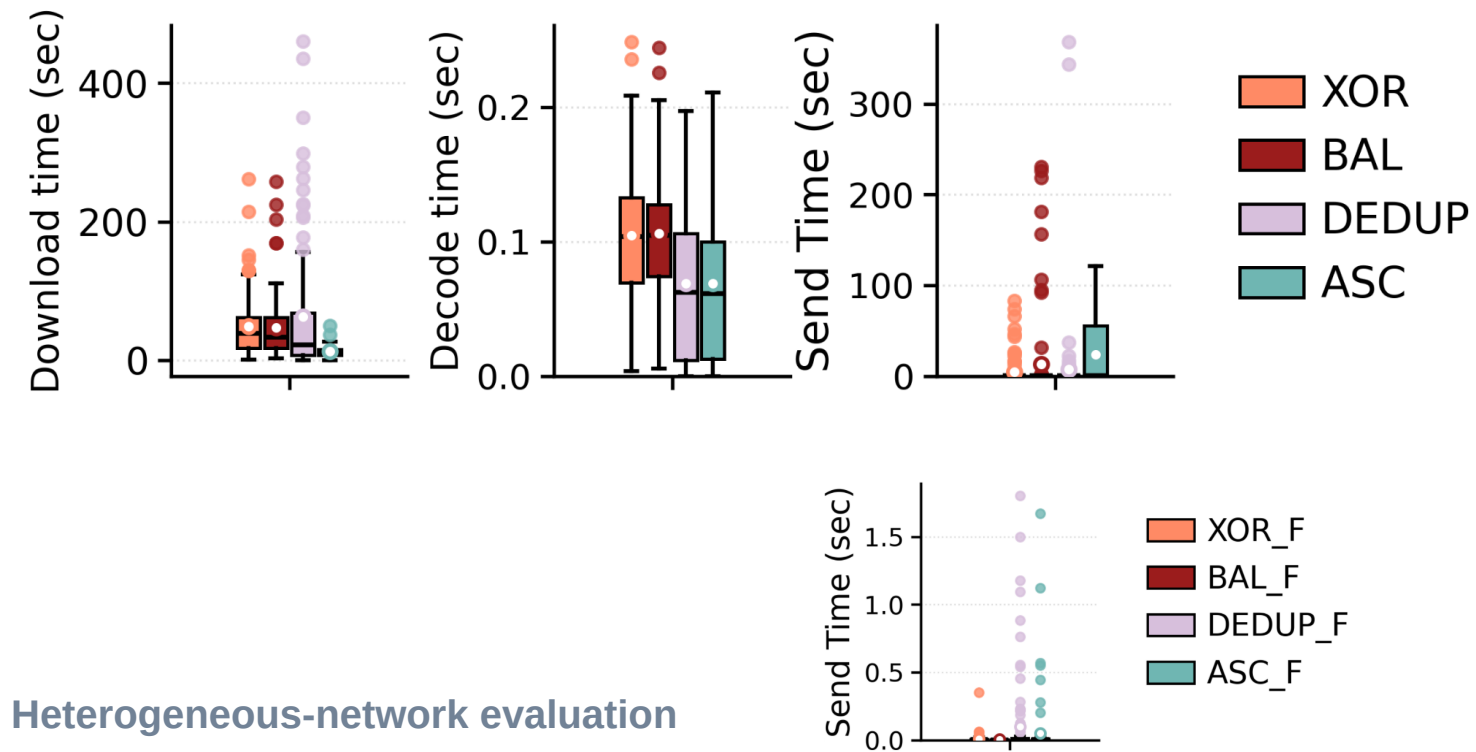
DOWNLOAD TIME
≈**96% lower**
than DEDUP

DOWNLOAD TIME
≈**92% lower**
than BAL and XOR

REMAINING LIMITATION
Send time
can dominate when the destination
node is slow.

ASC reduces download times but is not aware of placement policies in the system

Asclepius significantly reduces download time



Heterogeneous-network evaluation

DOWNLOAD TIME
≈96% lower
 than DEDUP

DOWNLOAD TIME
≈92% lower
 than BAL and XOR

REMAINING LIMITATION
Send time
 can dominate when the destination
 node is slow.

ASC reduces download times but is not aware of placement policies in the system

Scheduling itself adds little overhead

50 shards

5.2 GB repaired

4.2 s

similarity collection

0.25 s

Max–Min scheduling

Optimization Distributed similarity collection reduces 4.2 s to about **300 ms** .

Asclepius makes repair placement duplication- and heterogeneity-aware

1

Exploit duplication + bandwidth

Asclepius uses duplicated data chunks and repair-node download bandwidth to make better repair-node selections.

2

Implemented on Altocumulus

Asclepius is implemented and evaluated on Altocumulus, an erasure-coded decentralized storage system built on IPFS.

3

Reduce repair time

Asclepius reduces repair time by up to

10×

compared with the evaluated baselines.

Ongoing work Jointly select the **repair node** and the **destination node** to further optimize the repair process.

up to 80% less repair traffic

up to 10× faster repair

This work is part of the **Inria–Hive Alvearium** collaboration.

Thank you

Questions?

OPEN TO OPPORTUNITIES
Research & Engineering

✉ mohammadrizk7@gmail.com

✉ mohammad.rizk@inria.fr



