2018 USENIX Annual Technical Conference (ATC'18)
Boston, MA, USA, July 11-13, 2018

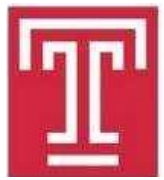
RAFI: Risk-Aware Failure Identification to Improve the RAS in Erasure-coded Data Centers

Juntao Fang †, Shenggang Wan†, and Xubin He§

†Huazhong University of Science and Technology, China

§Temple University, USA

华中科技大学
Huazhong University of Science & Technology

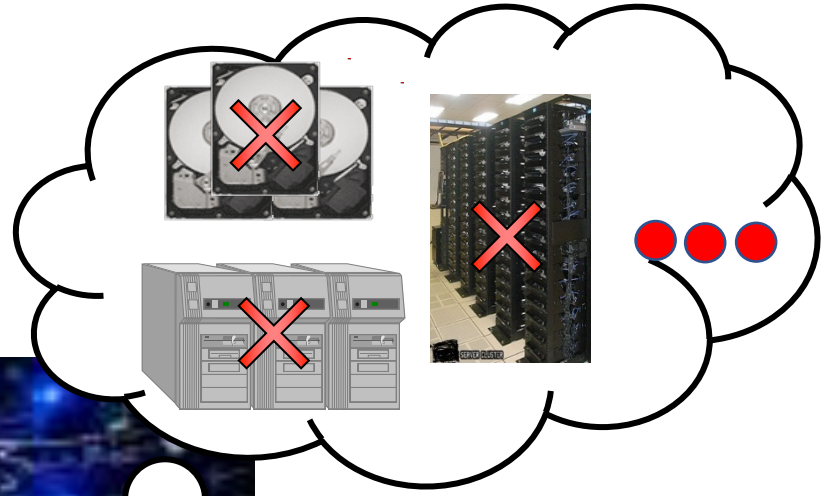


TEMPLE
UNIVERSITY®

Outline

- Background and Motivation
- RAFI Design
- Evaluations
- Conclusions

Storage in Data Centers



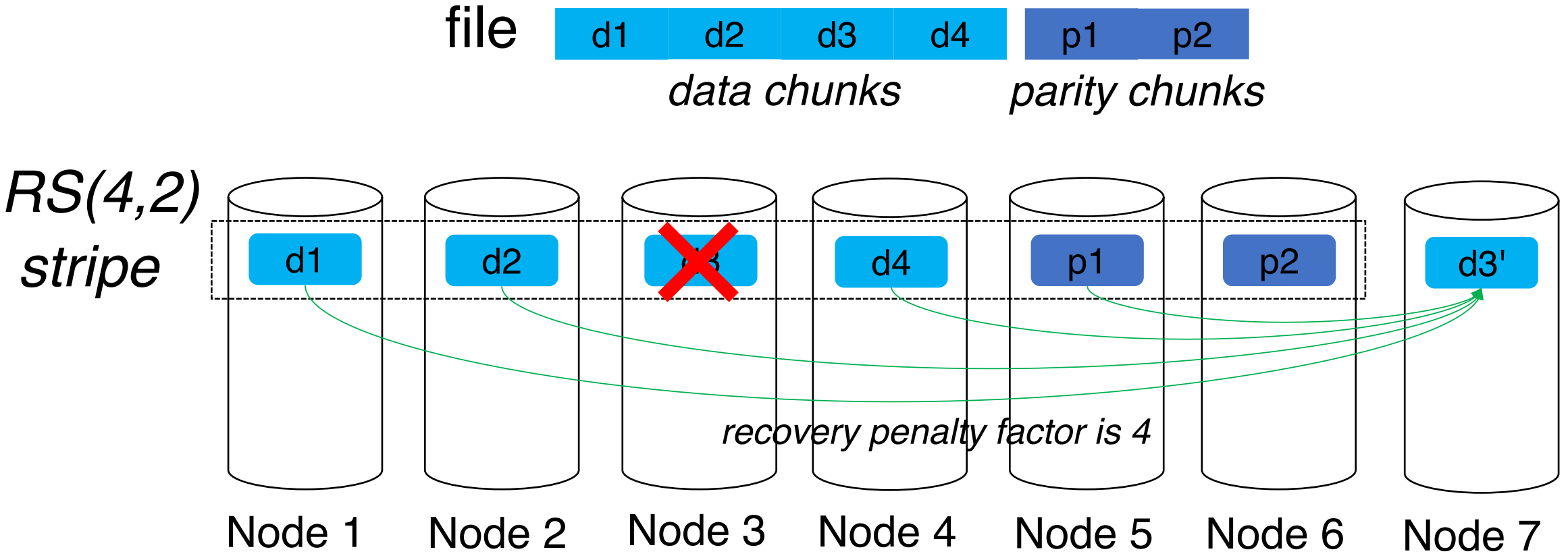
Reliability

Availability

Serviceability

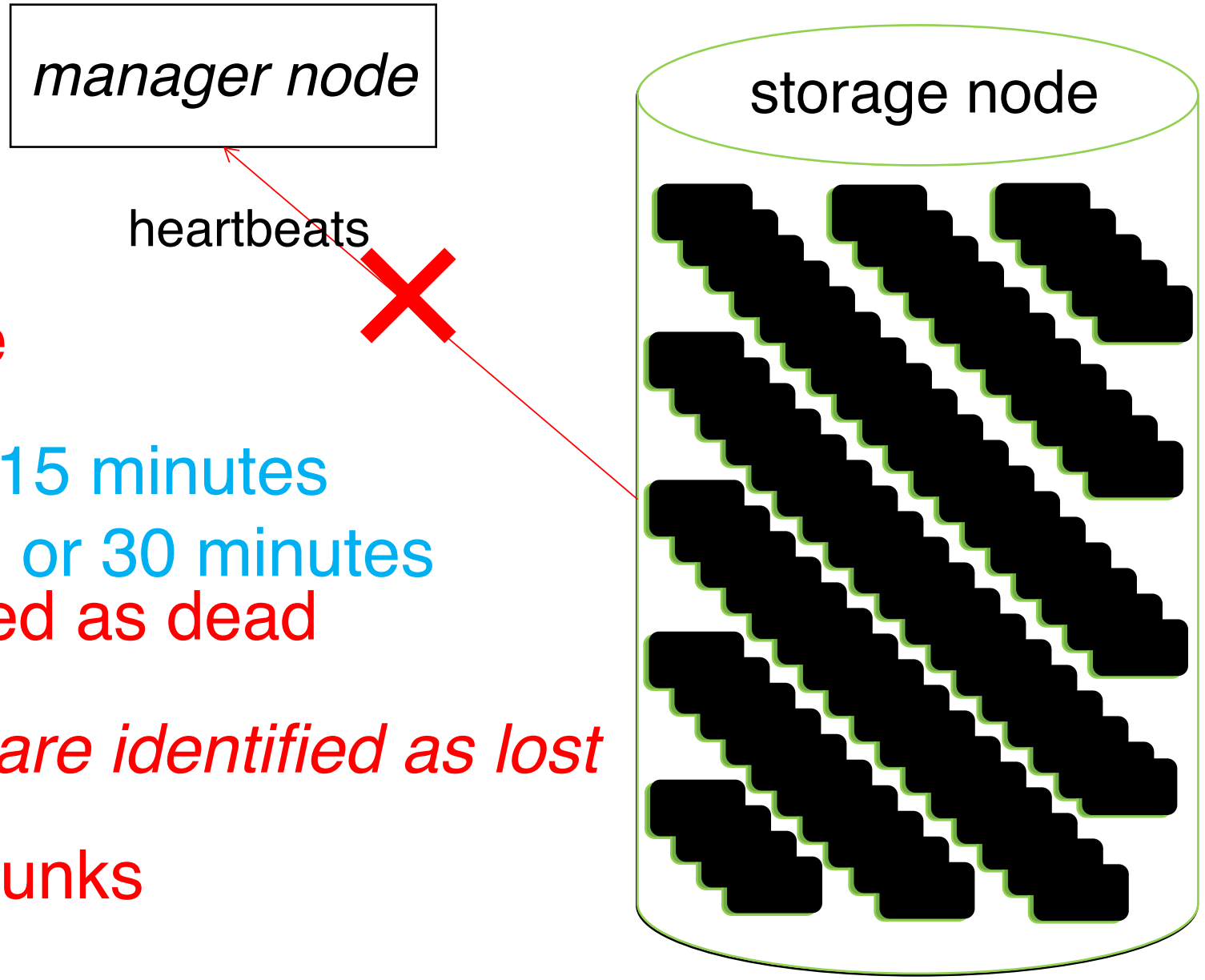
source: https://www.wsj.com/articles/why-data-centers-collect-big-tax-breaks-1416000057

Data Redundancy: Erasure Coding



high recovery penalty factor => high repair cost

Data Repair



Identification

1. Heartbeats lose
2. Time threshold:15 minutes
   or 30 minutes
3. Node is identified as dead

Recovery

4. *Lots of chunks are identified as lost*
5. Recover lost chunks

Reliability and Repair Cost

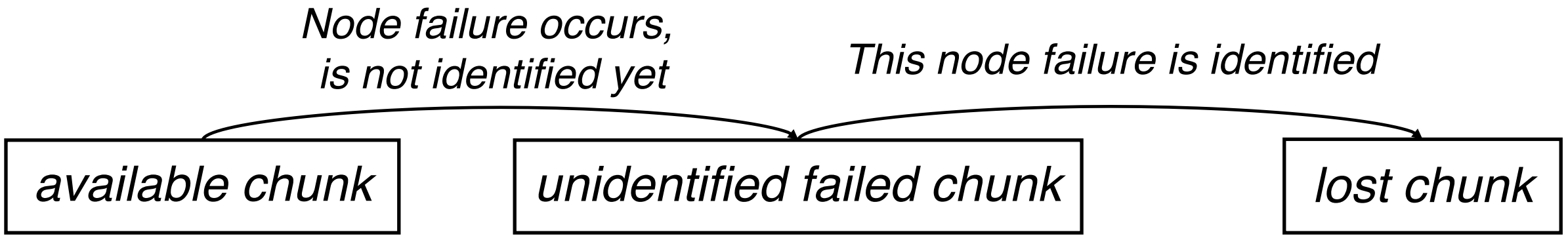
Mean Time to Data Loss: $$\mathrm{MTT}DL = \frac{\mathrm{MTTF}^{m+1}}{\prod_{j=0}^{m-1}(k+m-j)\times\prod_{i=1}^{m} MTTR_i}$$

Existing Methods

	Identification time	**Recovery time**		
	Time threshold ↓	Recovery penalty factor ↓ (LRC,MSR codes)	Recovery bandwidth ↑ (High speed network)	Queue Time ↓ (Priority)
Reliability	↑	↑	↑	↑
Repair cost	↑	↓	→	→

?

Problem Statement



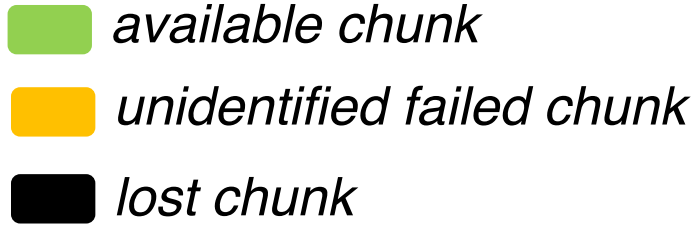
Identification of chunk failures ***relies on*** identification of node failures

We focus on the identification of chunk failures which is seldom studied.

Risk-Aware Failure Identification (RAFI)

Our solution: Identify chunk failures according to the risk level of their host stripes and apply different time thresholds accordingly.

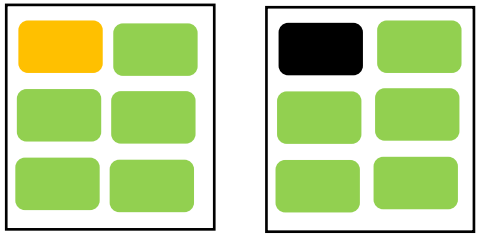
Stripes



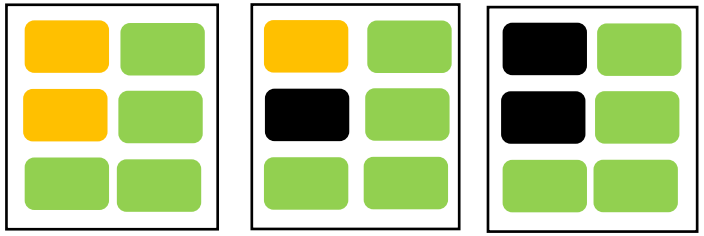
Risk level: the number of failed chunks

failed chunk: *unidentified failed chunk or lost chunk*

low risk stripe

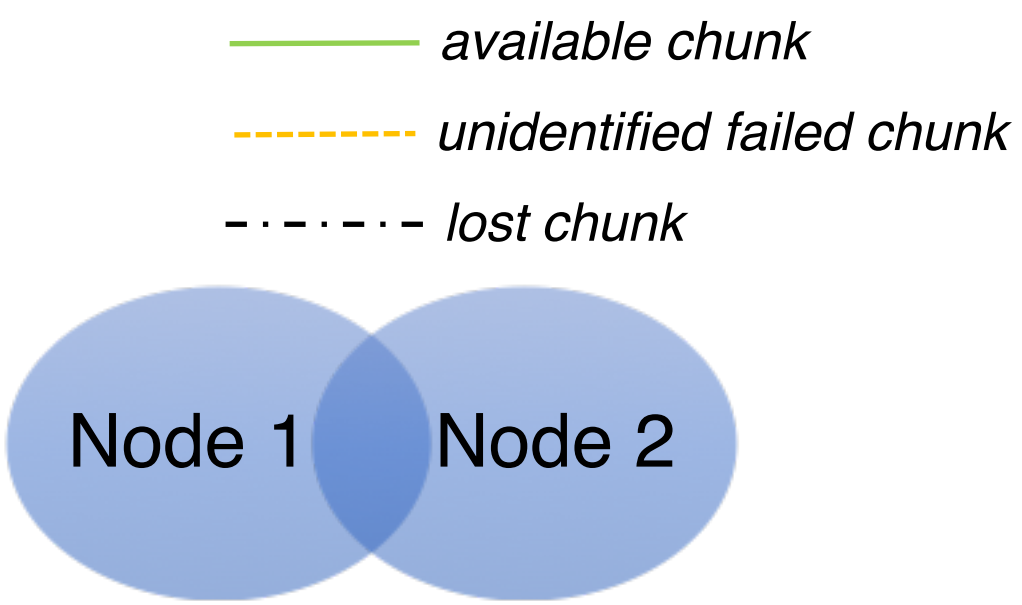


high risk stripe

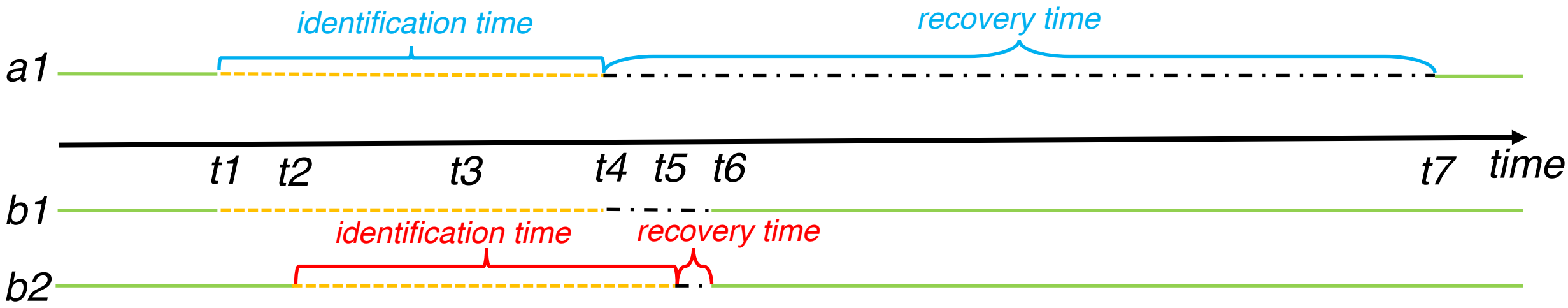


Observations

High risk stripes are far fewer than low risk stripes.



Failed chunks in low risk stripes



Failed chunks in high risk stripes

Identification of Chunk Failures

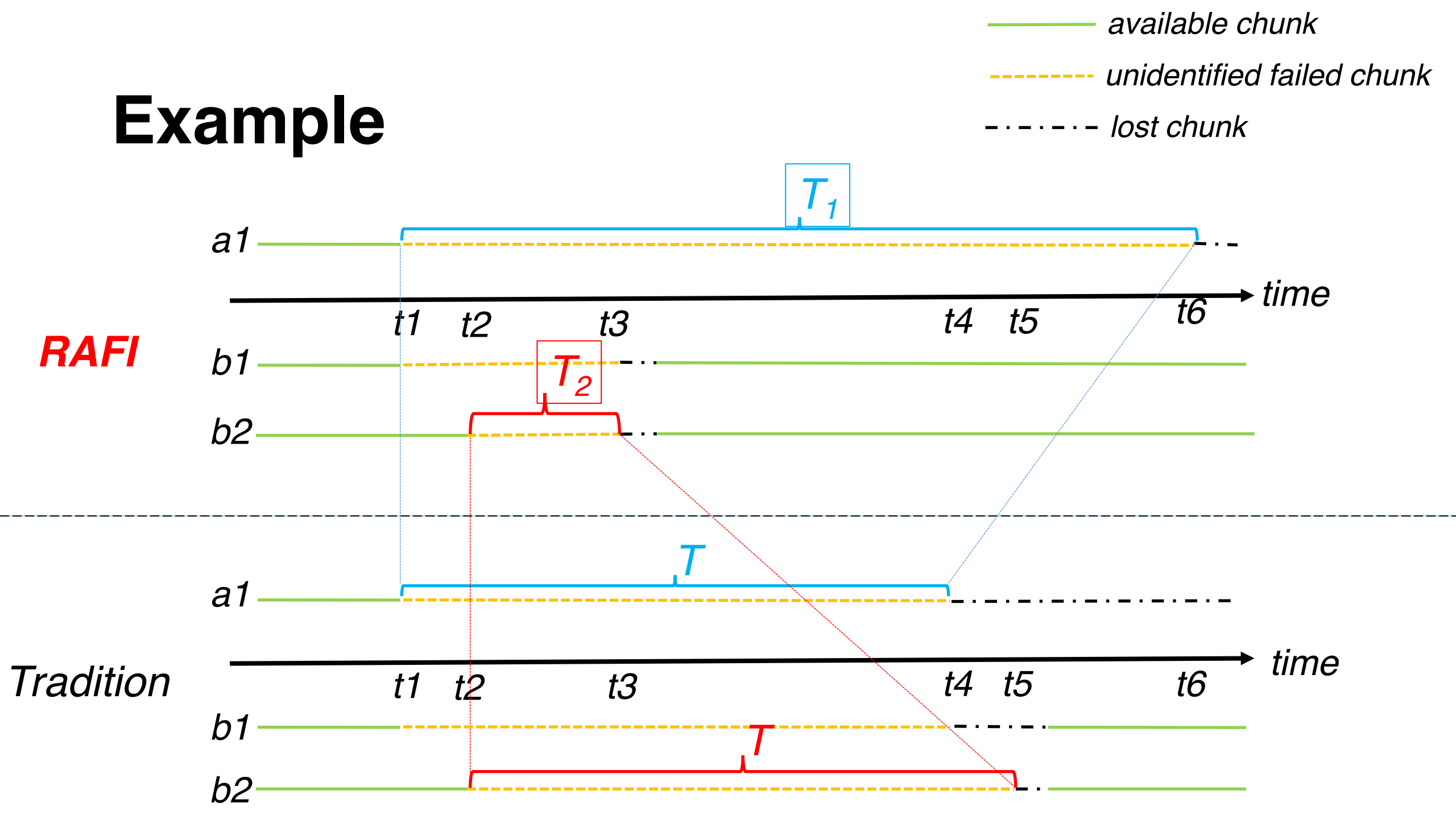
The more failed chunks a stripe has,
the shorter failure identification threshold those chunks take.

Failure occurs,
is not identified yet

$\exists$ i,
1. There are another i-1 failed chunks,
2. Failure durations of these i failed chunks are all longer than T_i

Preset time thresholds: T_i $(1 \leq i \leq m)$, T_i decreases as i increases

Example


available chunk
unidentified failed chunk
lost chunk
RAFI
T_1
a1
time
t1
t2
t3
t4
t5
t6
b1
T_2
b2
Tradition
T
a1
time
t1
t2
t3
t4
t5
t6
b1
T
b2

Identification of Chunk Failures

- *Different* time thresholds
  - *Each time threshold is set independently*
- Failed chunks in the stripe
  - # of failed chunks in the stripe
  - Failure durations of these failed chunks

Benefit and Cost

- Improving the RAS
  - All the time thresholds can be set independently to get proper trade-offs between the data reliability and availability, and the repair network traffic for a certain type of stripes.
- Compatibility
  - Work together with existing optimizations which focus on the failure recovery phase

- *Increasing degraded reads*
  - *Less than 1.7%*
- *Memory usage*
  - *Failed chunk lists -> failed node lists*

Evaluation

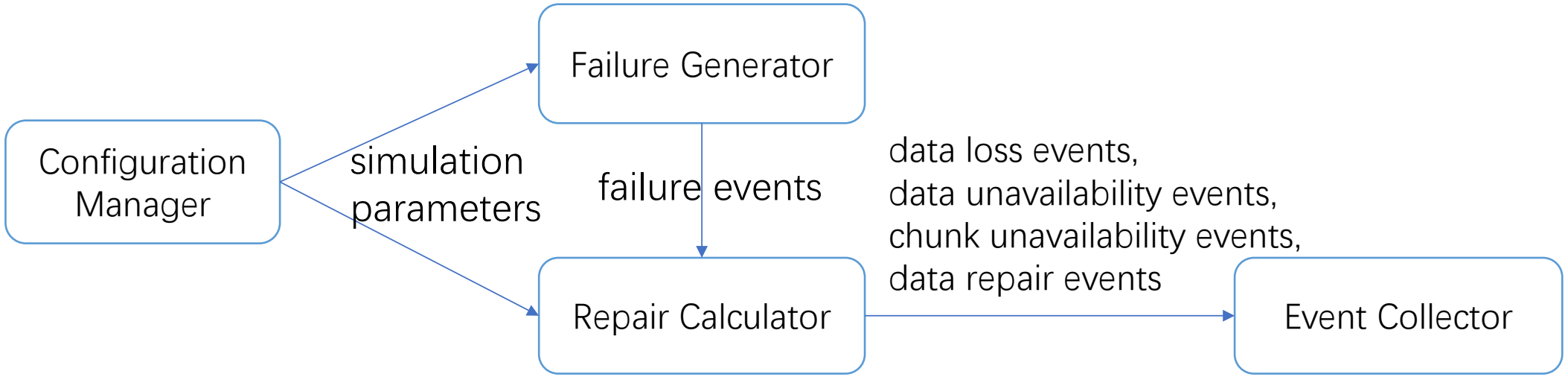
- Simulations + Prototype implementation
- The effectiveness and efficiency of RAFI on the RAS are evaluated through simulations.
- The design details and computational cost of RAFI are verified through prototyping running on a real distributed storage system.

Simulations

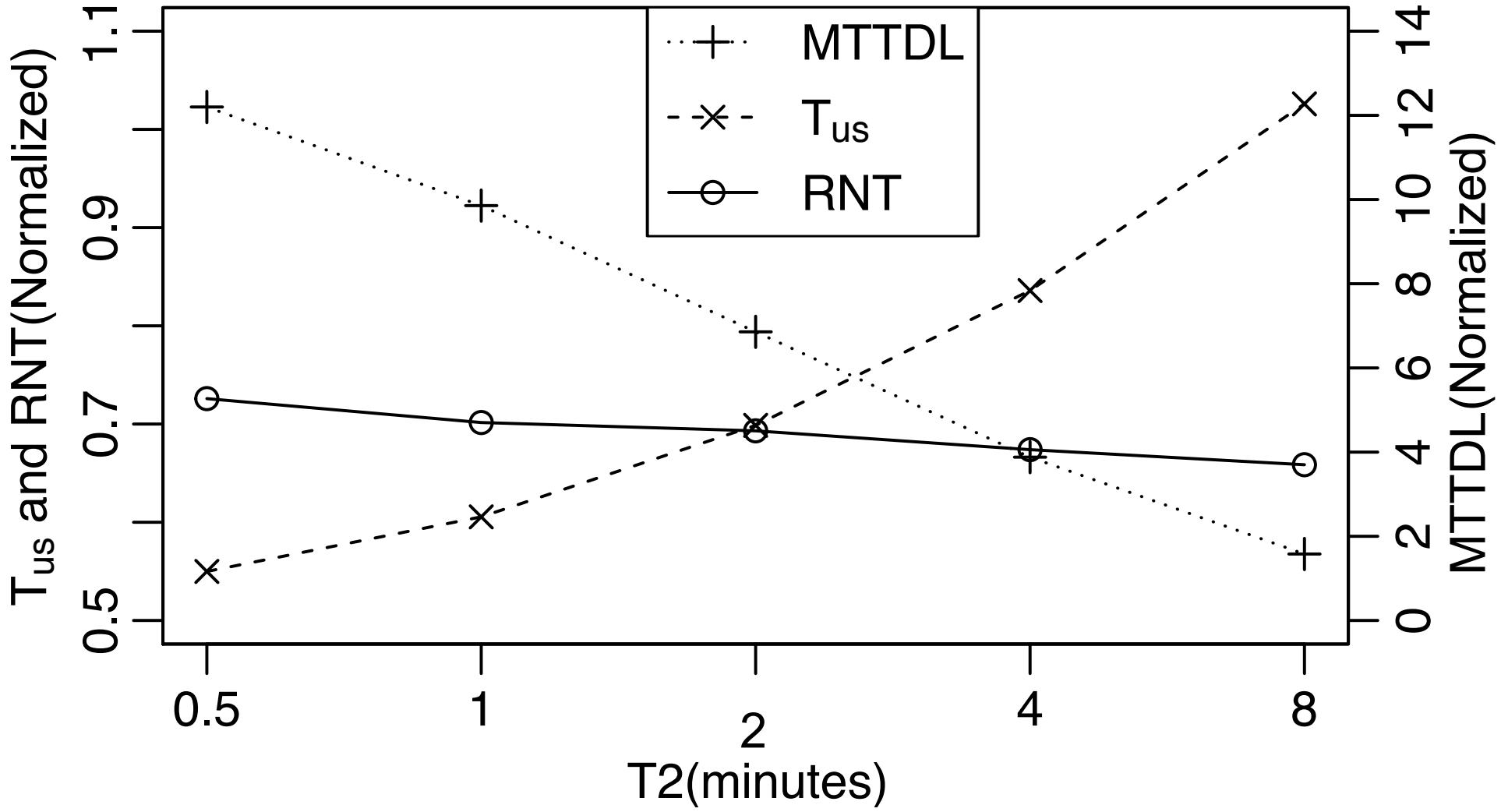
Symbol	Definition	Default Value
N	# of storage nodes in a data center	1000
d	# of chunks on a node	125,000
s	Chunk size	128 MB
T_h	Check interval of node states	5 minutes
b	Recovery network bandwidth on each node	0.1 Gbps
T_d	Duration of each iterations	5 years
N_i	# of iterations	500,000

DR-SIM

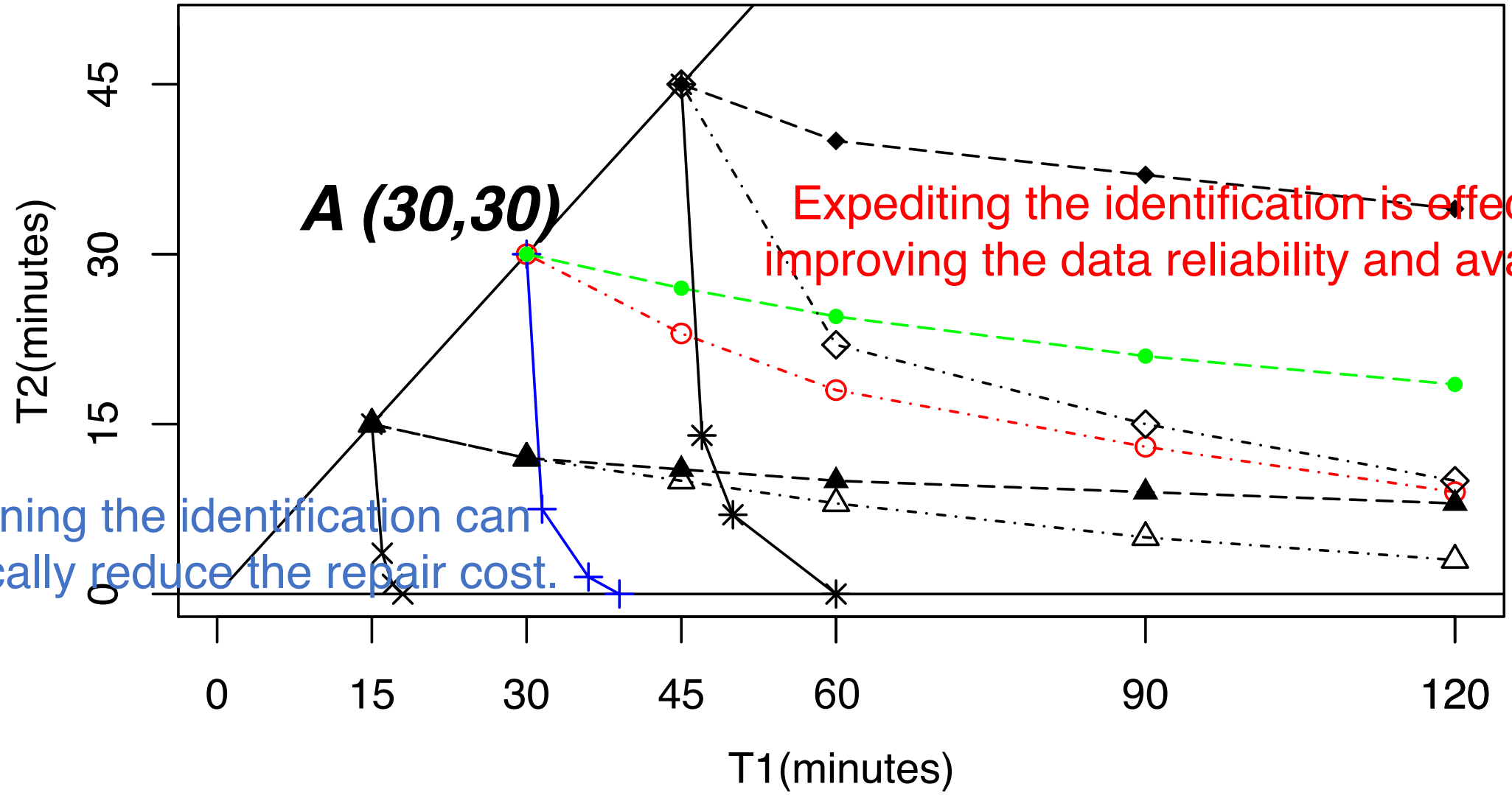
- Event-driven model
- Based on Monte Carlo Method



Improving the RAS



Improving the RAS: (T_1, T_2)



Summary of Simulations

- A simulator is developed to verify our RAFI
- Extensive simulations are conducted
  - Different time thresholds
  - Different kinds of erasure codes
  - Different network bandwidth
  - Compare with Lazy

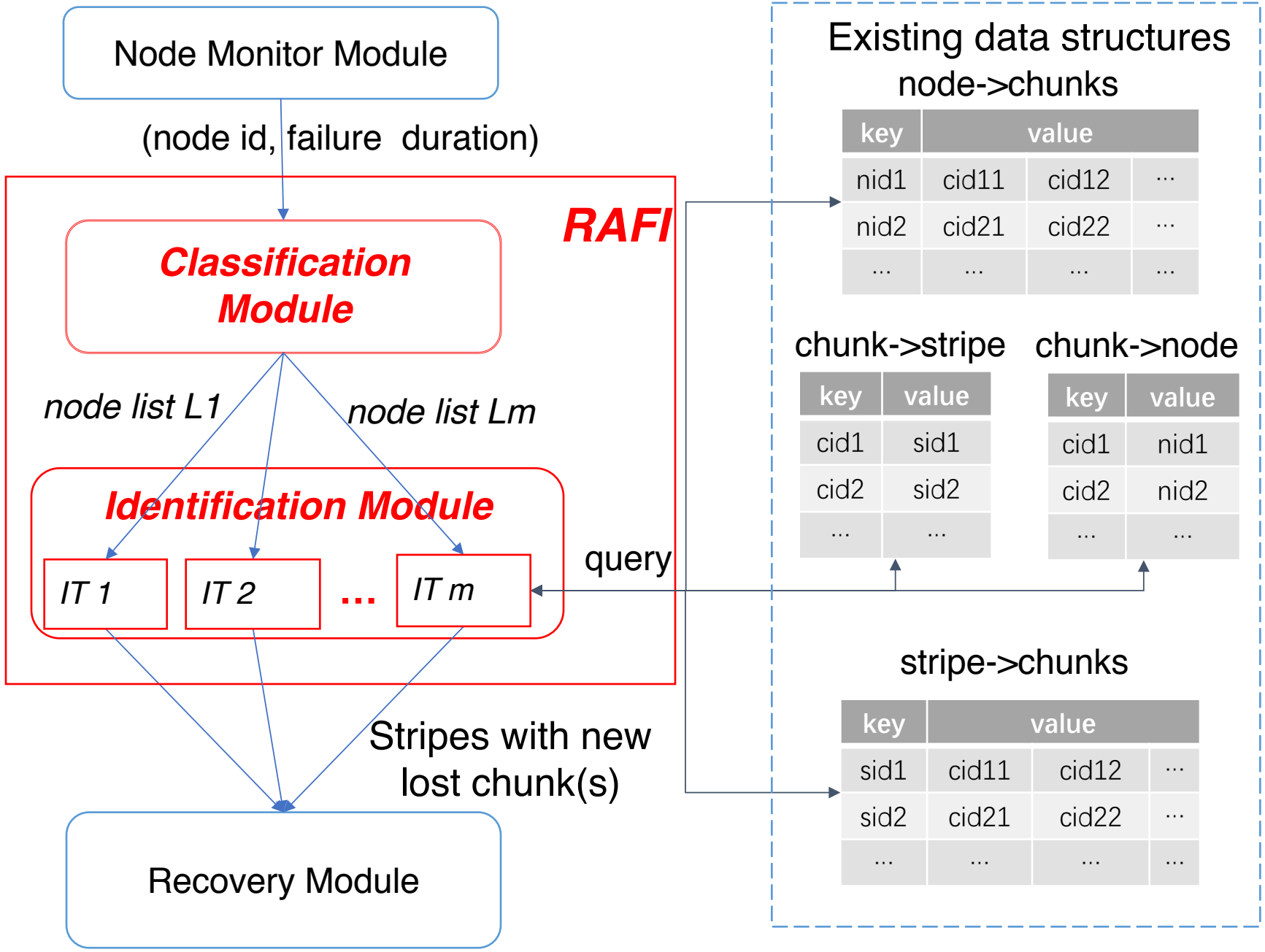
To further evaluate the effectiveness of RAFI, we use the prototyping experiments.

Implementation

RAFI-HDFS

- Based on HDFS 3.0.0-alpha2
- 200 loc

Memory usage
=>Computational Cost

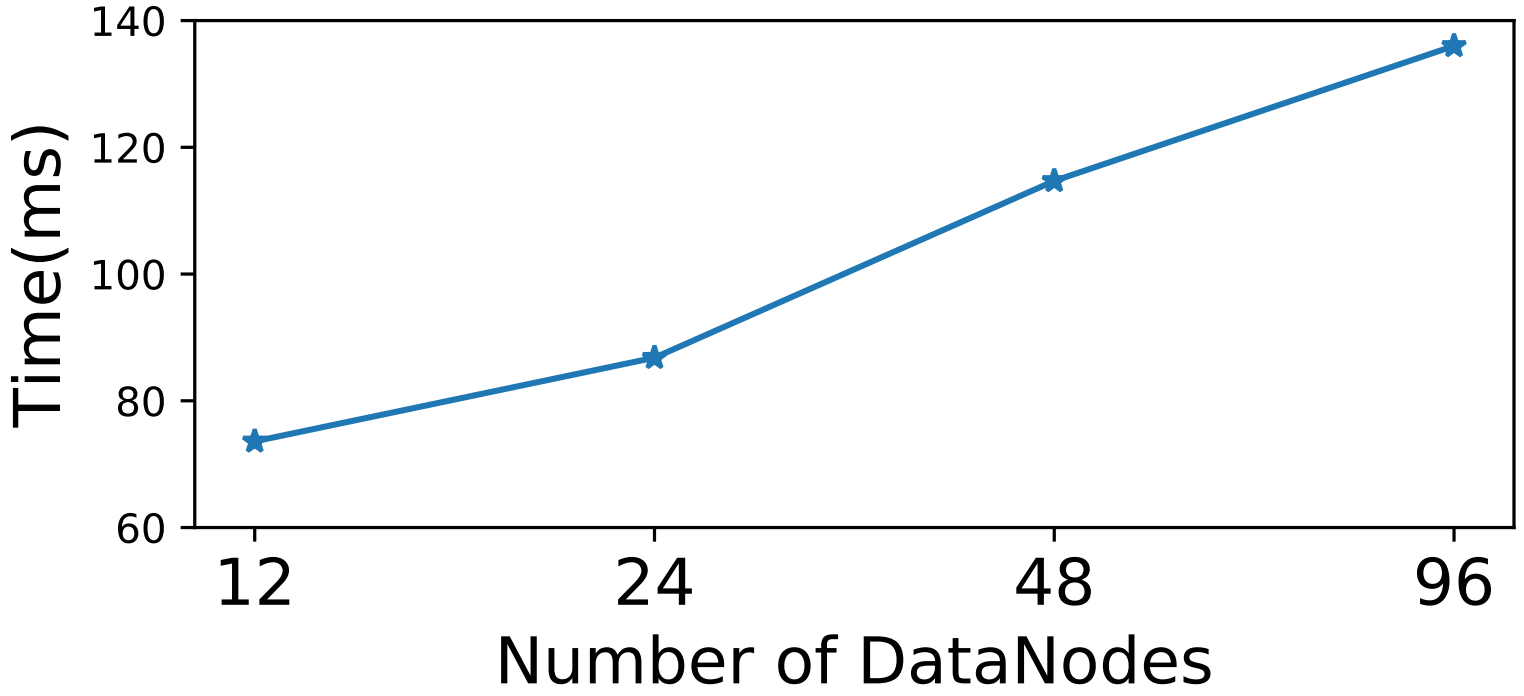


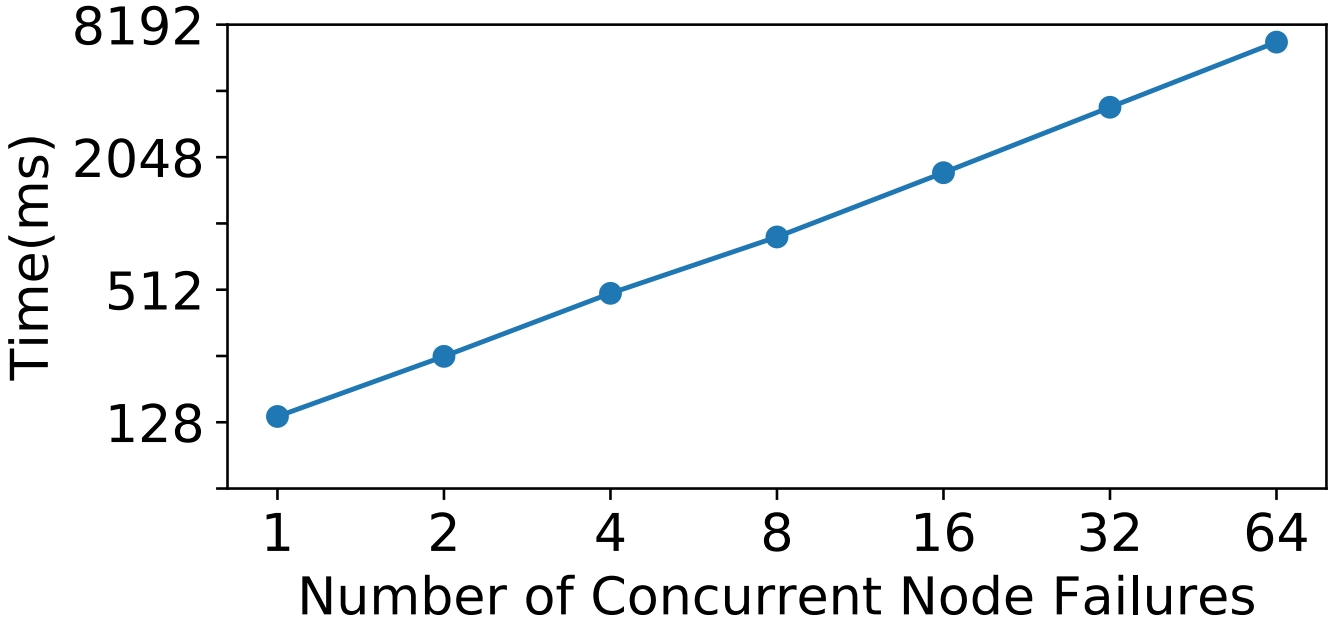
Prototyping Experiments

- Setups
  - 1 NameNode and 96 DataNodes
- Metrics
  - Identification time
  - Computational cost
    - System scale
    - Concurrent node failures

# storage nodes	96
CPU	Intel Xeon E5-2680v3 @ 2.5 GHZ (1 vCPU)
Memory	16 GB DDR4
Network	1 Gbps
OS	Ubuntu 14.04
HDFS	3.0.0-alpha2
# chunks on each storage nodes	68,000

Computational Cost





$\mathcal{O}$(# of failed chunks)

Conclusions

- We propose a risk-aware failure identification scheme RAFI to simultaneously improve the RAS
  - A chunk failure is identified through multiple independent identification thresholds based on their risk level of the stripes.
- A simulator is developed to verify our RAFI
  - RAFI can further improve the data reliability by a factor of 9.3, and reduce the data unavailability and repair network traffic by 43% and 36%, respectively, at the cost of degraded reads increased by 1.7%.
- A prototype of RAFI is implemented in HDFS
  - The computational cost of RAFI is negligible.

Acknowledgements

- Our shepherd, ***Dahlia Malkhi***, for her very detailed comments and helpful suggestions.
- The anonymous reviewers for their invaluable comments.

Thank you!

Questions?

Independently