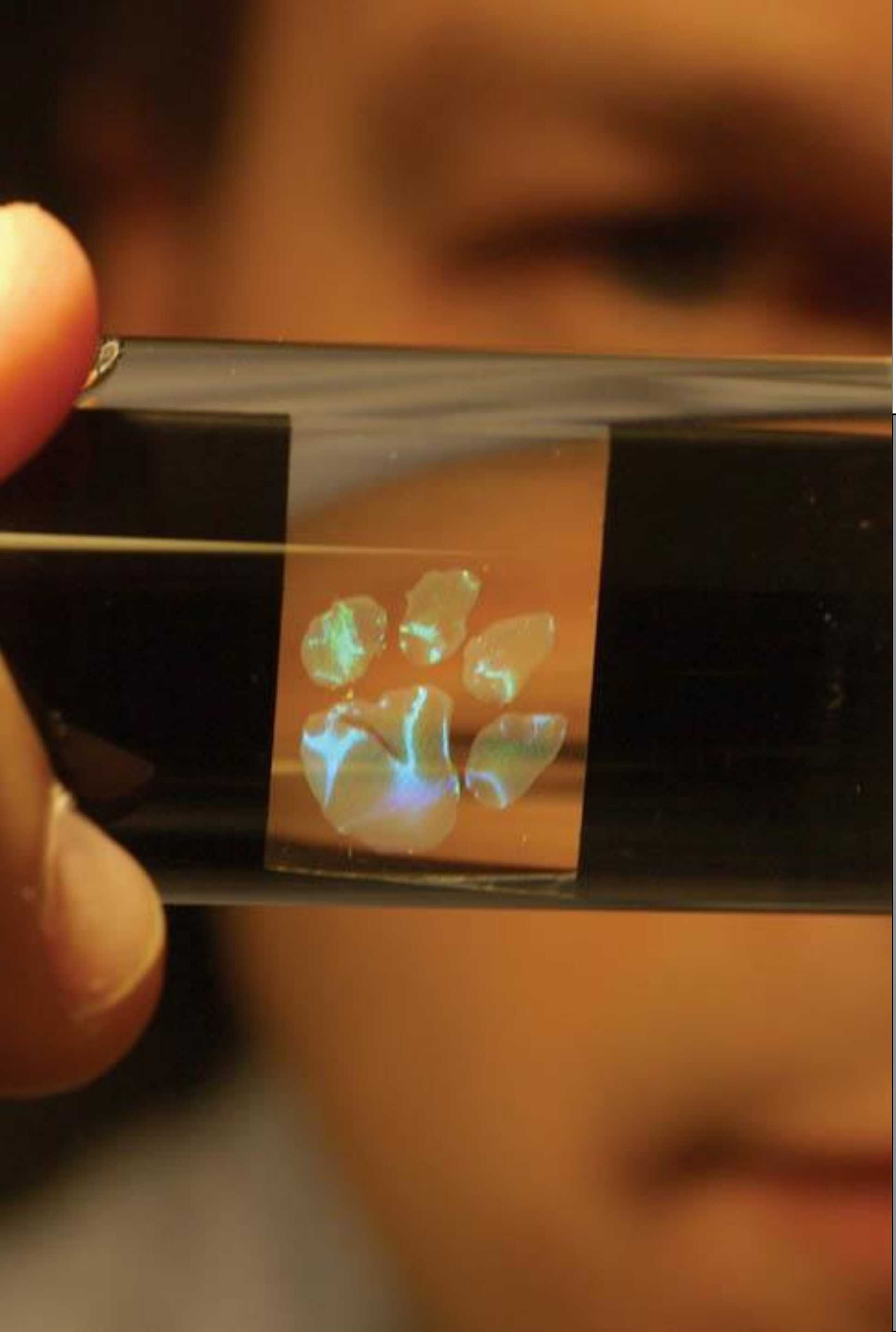


CLEMSON UNIVERSITY

UNIVERSITY of VIRGINIA

A Low-Cost Multi-Failure Resilient Replication Scheme for High Data Availability in Cloud Storage

Jinwei Liu* and Haiying Shen†

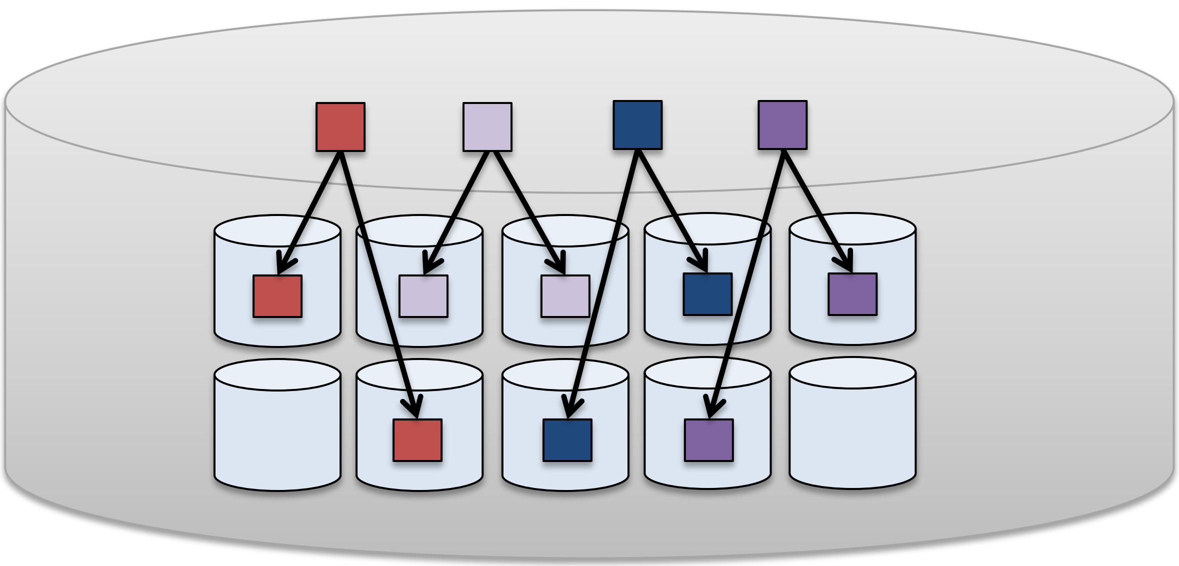
*Dept. of Electrical and Computer Engineering
Clemson University, SC, USA
†Dept. of Computer Science, University of Virginia, Charlottesville, VA, USA

Outline

- Introduction
- A Low-Cost Multi-Failure Resilient Replication Scheme (MRR)
- Design of MRR
- Performance Evaluation
- Conclusions

Introduction

- Data management in cloud storage

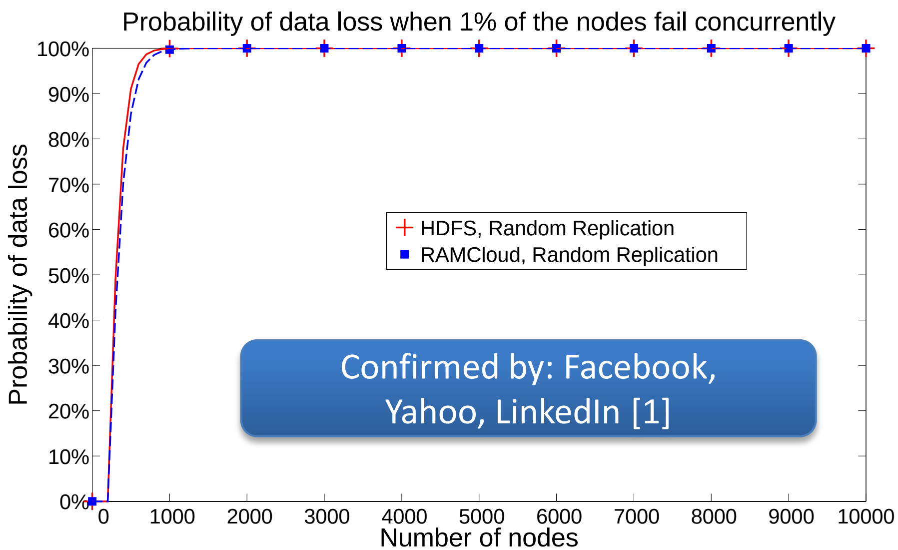


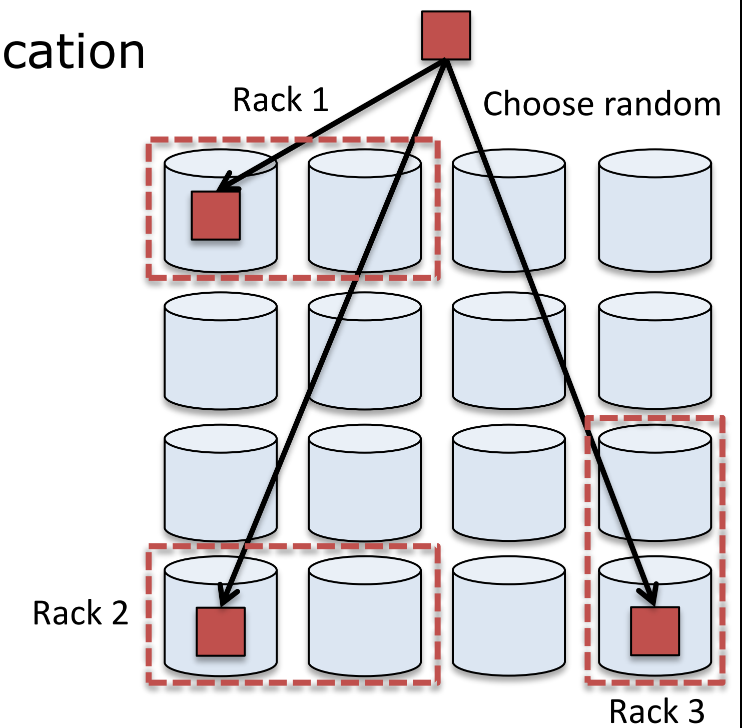
Motivation

- Data loss and machine failures in emerging cloud systems
  - Non-correlated machine failures
    - Multiple machines fail concurrently
  - Correlated machine failures
    - Machines fail individually
      - Power outages
        - » 1-2 times a year [Google, LinkedIn, Yahoo]
      - Large scale network failures
        - » 5-10 times a year [Google, LinkedIn]
      - And more
        - » Rolling software/hardware updates
- Design principle
  - Multi-failure resilient replication scheme

Motivation (cont.)

- Random Replication
  - Prob. of data loss in Random Replication



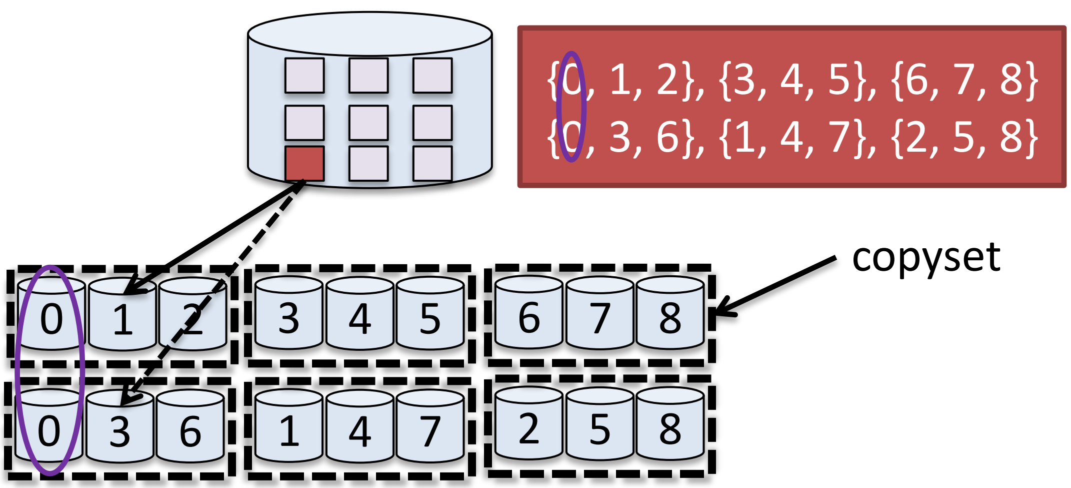


HDFS, GFS, Windows Azure, RAMCloud

[1] A. Cidon, S. Rumble, R. Stutsman, S. Katti, J. Ousterhout, and M. Rosenblum. Copysets: Reducing the frequency of data loss in cloud storage. In *Proc. of ATC*, 2013.

Motivation (cont.)

- Limitation of existing approaches
  - Random Replication
    - High data loss probability, high storage cost and consistency maintenance cost
  - Copyset Replication
    - High storage cost and consistency maintenance cost

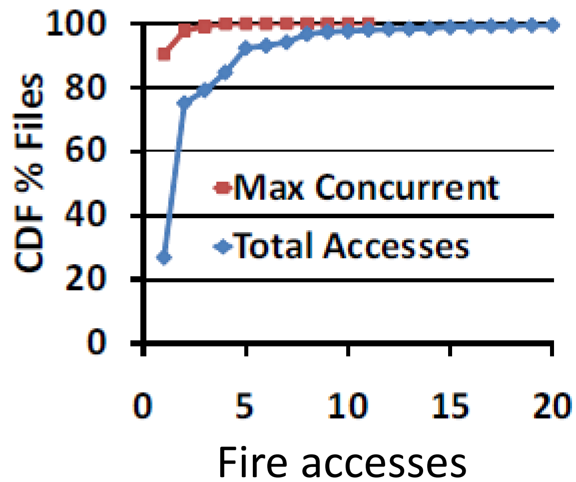


Scatter width (S): # of possible nodes storing the secondary replicas of a chunk

- Design principle
  - Cost-effective replication scheme

Motivation (cont.)

- Data popularity existing in cloud storage systems [2-3]
  - File popularity
    - CDFs of the total # of jobs that access each file and the # of concurrent accesses [2]



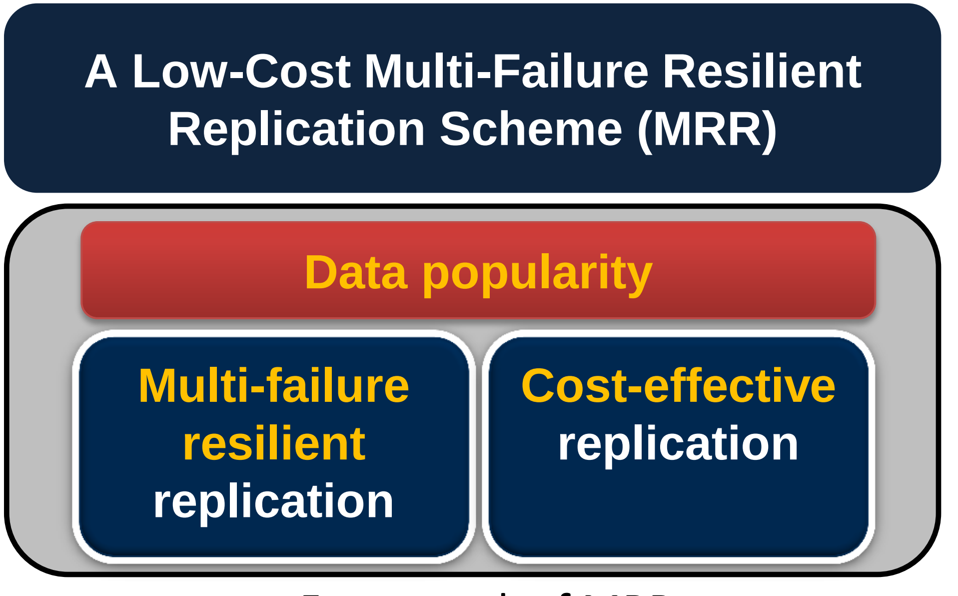
- Design principle
  - Popularity-aware replication

[2] G. Ananthanarayanan, S. Agarwal, S. Kandula, A. Greenberg, I. Stoica, D. Harlan, and E. Harris. Scarlett: Coping with skewed content popularity in mapreduce clusters. In *Proc. of EuroSys*, 2011.

[3] A. Khandelwal, R. Agarwal, and I. Stoica. BlowFish: Dynamic Storage-Performance Tradeoff in Data Stores. In *Proc. of NSDI*, 2016.

Proposed Solution

- MRR: A Low-Cost Multi-Failure Resilient Replication Scheme
  - Features of MRR
    - Popularity awareness
    - Multi-failure resilience
    - Cost-effectiveness



Framework of MRR

Outline

- Introduction
- A Low-Cost Multi-Failure Resilient Replication Scheme (MRR)
- Design of MRR
- Performance Evaluation
- Conclusions

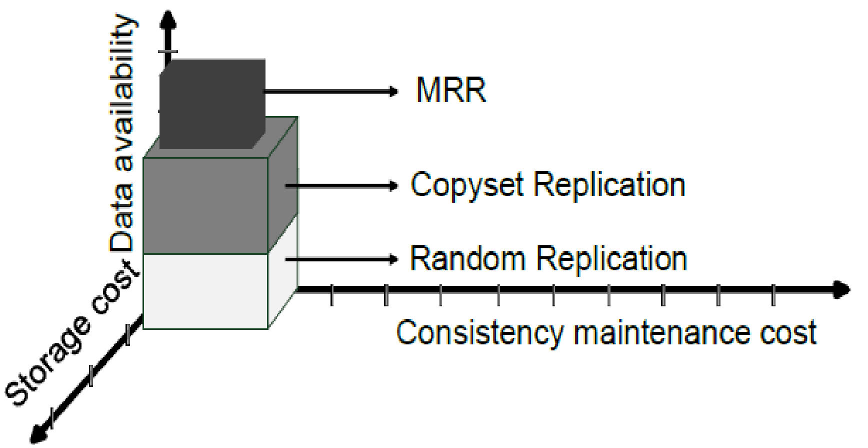
MRR

- Concepts
  - Correlated machine failures: multiple machines (servers) fail simultaneously
  - Non-correlated machine failures: machines fail individually
  - Fault-tolerant set (FTS): a distinct set of servers holding all replicas of a given data chunk

- Problem statement
  - Replicate the chunks of data objects so that the request failure probability, storage cost and consistency maintenance cost are minimized in both correlated failures and non-correlated failures

MRR

- Goal
  - Design a low-cost multi-failure resilient replication scheme for achieving high data availability while reducing storage cost and consistency maintenance cost caused by replication



Challenges

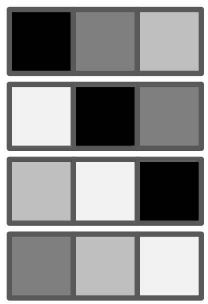
- Challenges of MRR design
  - How to significantly reduce data loss probability in both correlated and non-correlated machine failures
  - How to leverage data popularity to reduce cost (storage cost and consistency maintenance cost) caused by replication without compromising expected data availability much
  - How to determine popularity of data objects and the replication degree of each data object

Outline

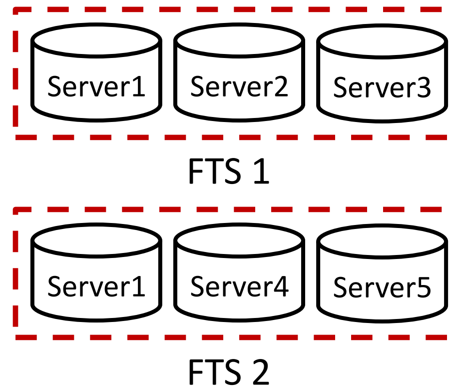
- Introduction
- A Low-Cost Multi-Failure Resilient Replication Scheme (MRR)
- Design of MRR
- Performance Evaluation
- Conclusions

Design of MRR

- Reduce data loss probability
  - BIBD-based method to generate FTSs (fault-tolerant sets) and constrain the replicas of each data chunk in one FTS



Balanced Incomplete Block Design (BIBD)



- Reduce cost
  - Use less replicas for unpopular data
  - Choose storage mediums for data objects based on data popularity

Data Popularity

- Determine data popularity
  - The popularity φ_{ij} of a data object (D_{ij}) is determined by its application rank and expected visit frequency (denoted by v_{ij}), i.e., # of visits in an epoch (say epoch t)

$$\varphi_{ij}(\cdot) = \alpha \cdot b_{a_i} + \beta \cdot v_{ij} \qquad (1)$$

  - where α and β are weights. The request probability of D_{ij} is proportional to its popularity, that is

$$r_{ij} = k_1 \cdot \varphi_{ij}(\cdot) \qquad (2)$$

  - where k_1 is a certain coefficient

Nonlinear Integer Programming Model

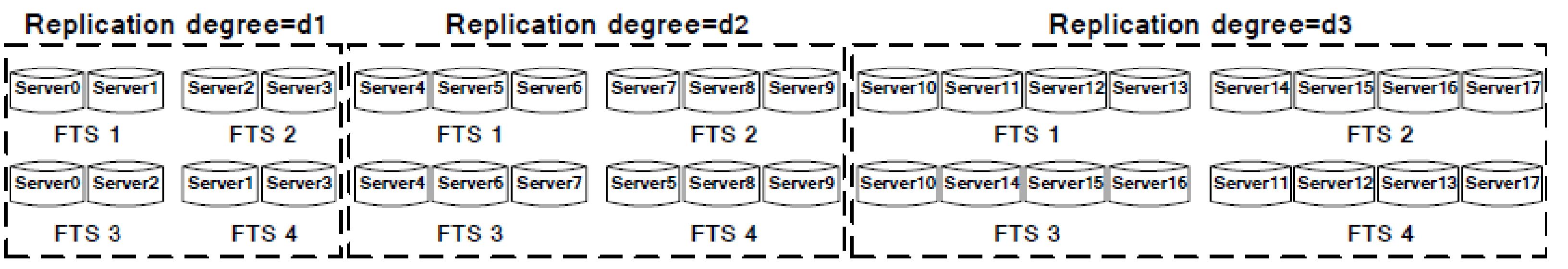
- Determine replication degree of data objects

$$min \ \ \{\bar{P}_r + C_c + C_s\} = \sum_{i=1}^{n}\sum_{j=1}^{m}(r_{ij} \cdot M \cdot (P_f)^{d_{ij}})$$
$$+ \sum_{i=1}^{n}\sum_{j=1}^{m}(M \cdot d_{ij}) \cdot \delta_{com} + \sum_{i=1}^{n}\sum_{j=1}^{m}(s_{ij} \cdot d_{ij} \cdot c_{s_{ij}} \cdot T)$$
$$s.t. \ \ \sum_{j=1}^{m} s_{ij} \cdot d_{ij} \leq K_i \ \ (i = 1, ..., n)$$
$$\sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij} \cdot M \cdot (P_f)^{d_{ij}} \leq P_r^{th} \ \ (0 < r_{ij} < 1)$$
$$\sum_{i=1}^{n}\sum_{j=1}^{m}(M \cdot d_{ij}) \cdot \delta_{com} \leq C_c^{th}$$
$$\sum_{i=1}^{n}\sum_{j=1}^{m}(s_{ij} \cdot d_{ij} \cdot c_{s_{ij}} \cdot T) \leq C_s^{th}$$

  - Relaxed NLIP optimization model is convex
  - Lagrange multipliers for deriving the solution for real-number optimization problem

System Design

- MRR algorithm



Architecture of MRR

  - Rank the replication degrees in ascending order $d_1, \ldots, d_l$
  - Group data objects with d_i $(i = 1, \ldots, l)$ together (D_i)
  - Use BIBD-based method to generate FTSs
  - Store each chunk's replicas with d_i to all nodes in an FTS with d_i

Outline

- Introduction
- A Low-Cost Multi-Failure Resilient Replication Scheme (MRR)
- Design of MRR
- Performance Evaluation
- Conclusions

Performance Evaluation

- Methods for comparison
  - Random replication (RR)

    Choose secondary replica holders from a window of nodes around the primary node based on Facebook's design

  - Copyset Replication (Copyset) [1]

    [1] A. Cidon, S. Rumble, R. Stutsman, S. Katti, J. Ousterhout, and M. Rosenblum. Copysets: Reducing the frequency of data loss in cloud storage. In *Proc. of ATC*, 2013.

  - Replication Degree Customization (RDC) [4]

    [4] M. Zhong, K. Shen, and J. Seiferas. Replication degree customization for high availability. In *Proc. of EuroSys*, 2008.

Experiment Setup

- Set parameters in Facebook, HDFS and RAMCloud environments

System	Chunks per node	Cluster size	Scatter width
Facebook	10000	1000-5000	10
HDFS	10000	100-10000	200
RAMCloud	8000	100-10000	N-1

- Distribution of file popularity and updates follow those of CTH trace
- Use CTH trace to generate data request
- 7 simulated data centers

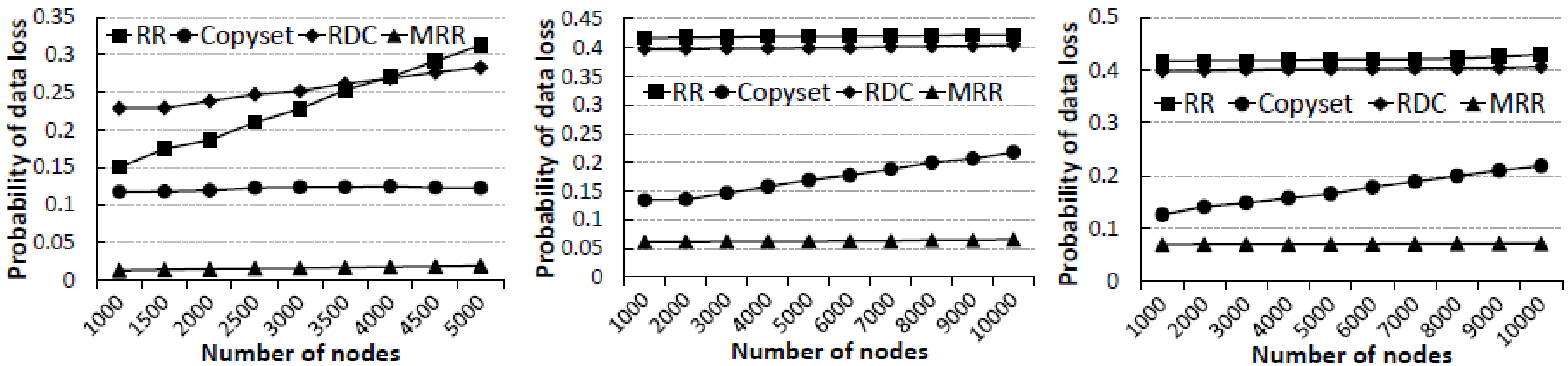
Experiment Setup (cont.)

- Parameter settings

Parameter	Meaning	Setting
N	# of servers	1000-10000
M	# of chunks of a data object	3
R	# of servers in each FTS	3
λ	# of FTSs containing a pair of servers	1
S	Scatter width	4
p	Prob. of a server failure	0.5
P_r^{th}	Threshold for expected request failure	0.05
C_c^{th}	Threshold for consistency maint. cost	1000000
C_s^{th}	Threshold for storage cost	300000
m	# of data objects in each application	1000
n	# of data applications	5

Evaluation of MRR

- Numerical analysis

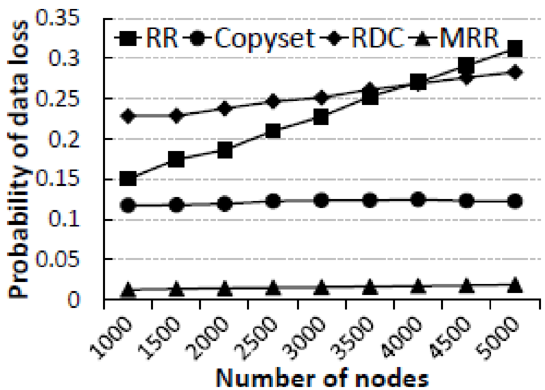


(a) Facebook (b) HDFS (c) RAMCloud

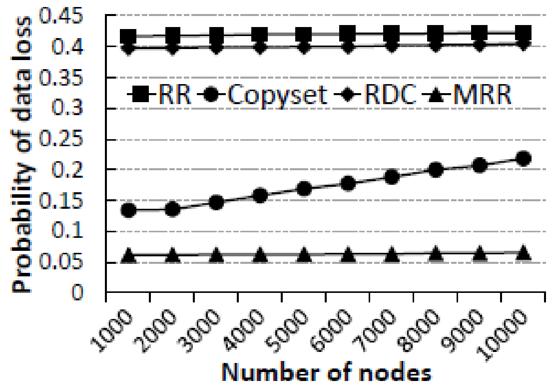
Prob. of data loss: MRR < Copyset < RDC < RR

Evaluation of MRR (cont.)

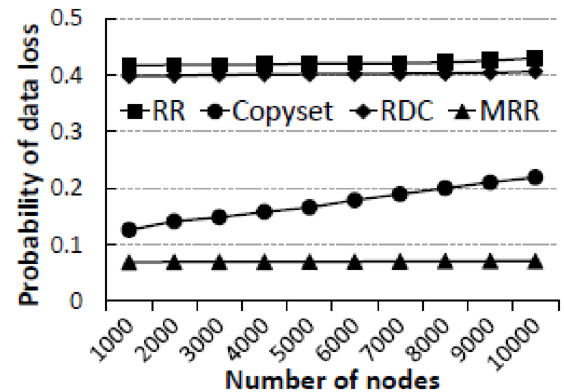
- Numerical analysis



(a) Facebook



(b) HDFS

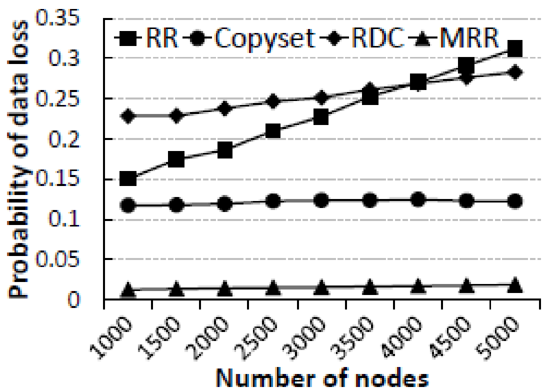


(c) RAMCloud

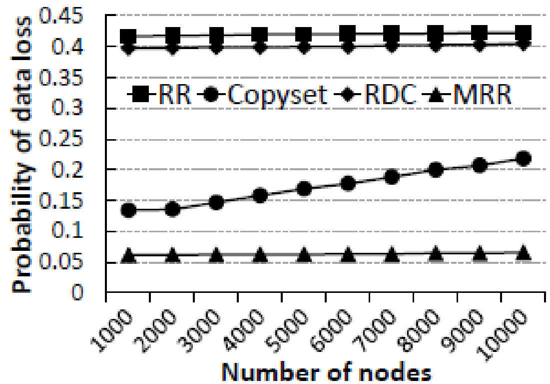
Availability: MRR > Copyset > RDC > RR

Evaluation of MRR (cont.)

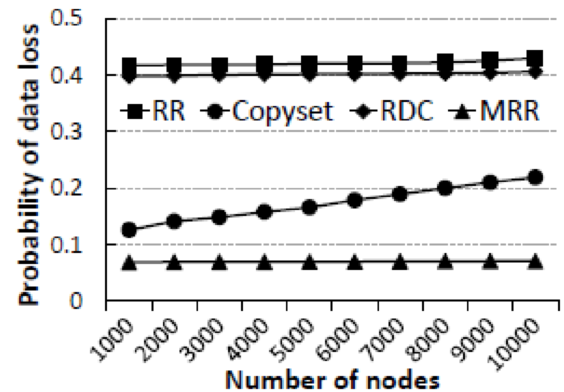
- Numerical analysis



(a) Facebook



(b) HDFS

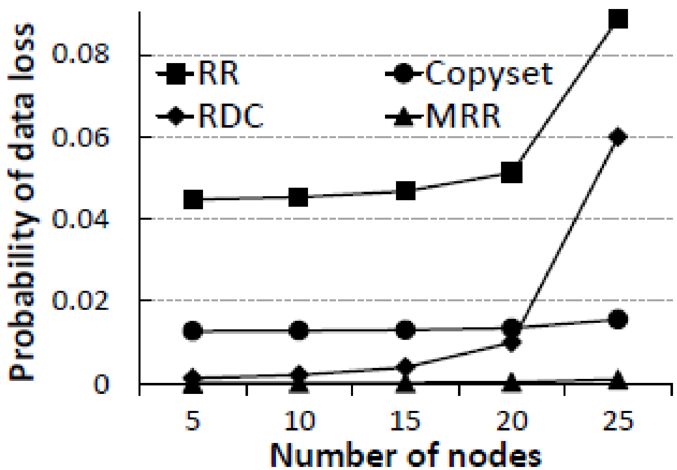


(c) RAMCloud

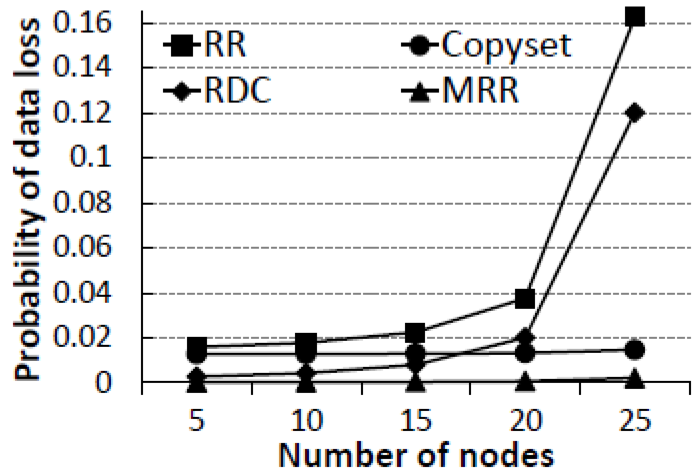
Result: Storage cost follows RR ≈ Copyset > RDC > MRR; consistency maintenance cost follows MRR < Copyset ≈ RR < RDC

Evaluation of MRR (cont.)

- Experimental results on Amazon S3



(a) Scatter width = 2

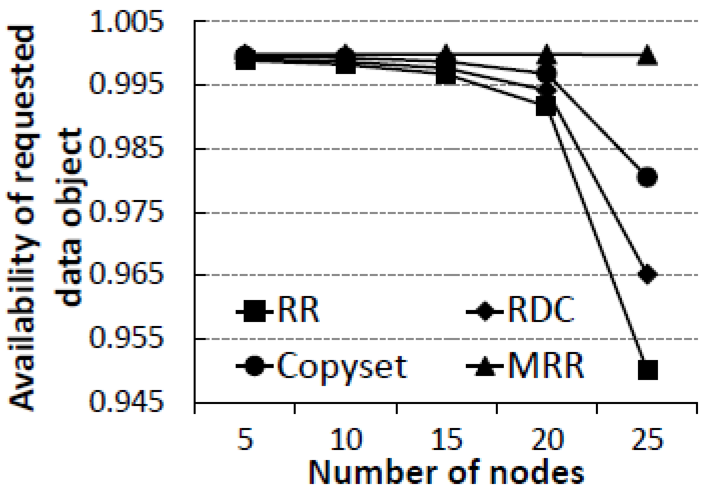


(b) Scatter width = 4

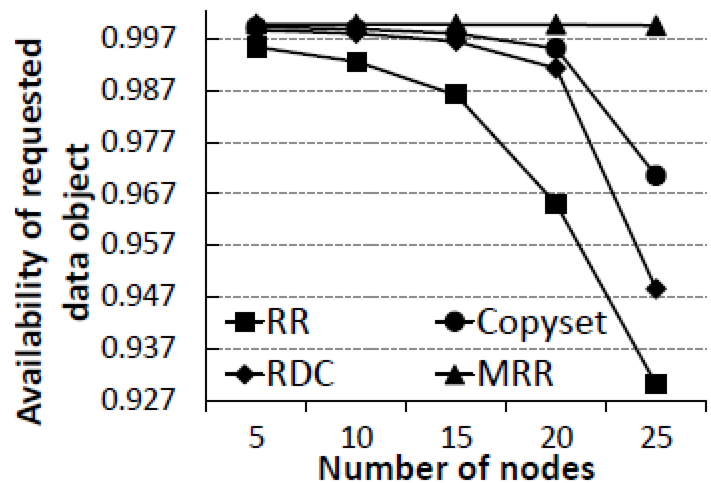
Prob. of data loss: MRR < RDC < Copyset < RR; prob. of data loss decreases as scatter width decreases

Evaluation of MRR (cont.)

- Experimental results on Amazon S3



(a) Scatter width = 2

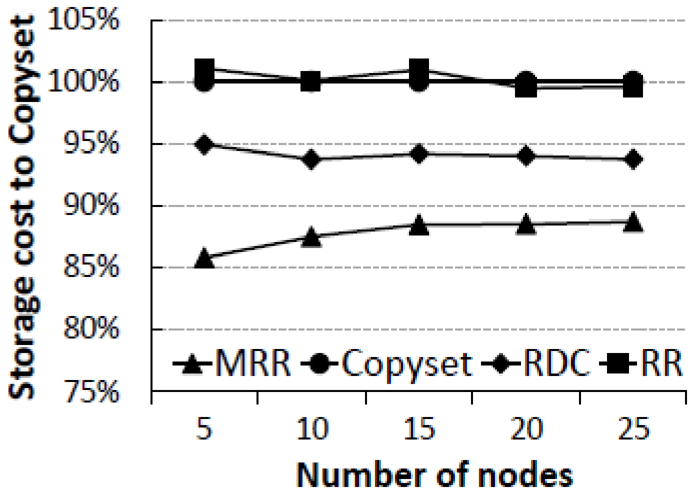


(b) Scatter width = 4

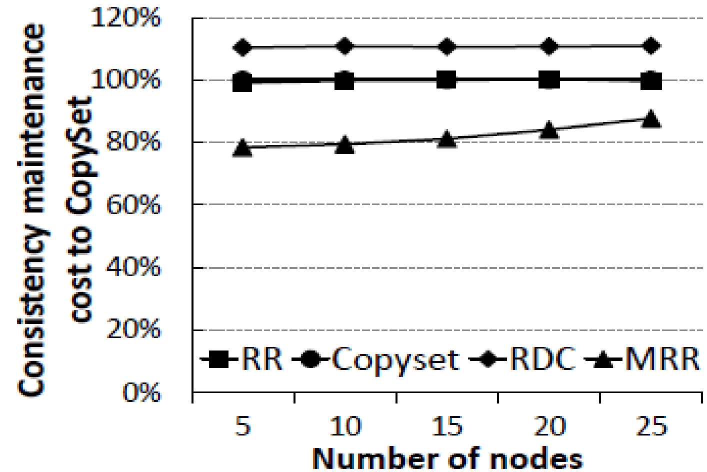
Availability: MRR > Copyset > RDC > RR; availability increases as scatter width decreases

Evaluation of MRR (cont.)

- Experimental results on Amazon S3



(a) Storage cost



(b) Consistency maintenance cost

Result: Storage cost ratio follows RR ≈ Copyset > RDC > MRR; consistency maintenance cost ratio follows RDC > RR ≈ Copyset > MRR

Outline

- Introduction
- A Low-Cost Multi-Failure Resilient Replication Scheme (MRR)
- Design of MRR
- Performance Evaluation
- Conclusions

Conclusions

- Our contributions
  - Build a NLIP model to maximize expected data availability with considering data popularity and reduce cost caused by replication
  - Based on the derived replication degree from NLIP, present MRR to handle data loss in correlated and non-correlated failures; MRR restricts replicas of a data chunk into an FTS, which reduces data loss probability
  - MRR uses different storage mediums for data objects based on data popularity to further reduce storage cost
  - Conduct extensive trace-driven experiments to compare MRR with other state-of-the-art replication schemes

- Future work
  - Update frequency for reducing consistency maintenance cost
  - Node joining and node leaving
  - Influence of changing network connections
  - Power consumption of machines

Thank you!

Questions & Comments?

Jinwei Liu, PhD

jinweil@clemson.edu

Electrical and Computer Engineering

Clemson University