

# Midterm Project – Chord

Practices for Distributed Systems and Cloud Application  
Development

# Goal

- Build a distributed file system based on Chord DHT
  - The system should automatically scale based on storage usage
  - The system should evenly distribute the load across all nodes
  - The system should be able to recover from failures (i.e. be fault-tolerant)

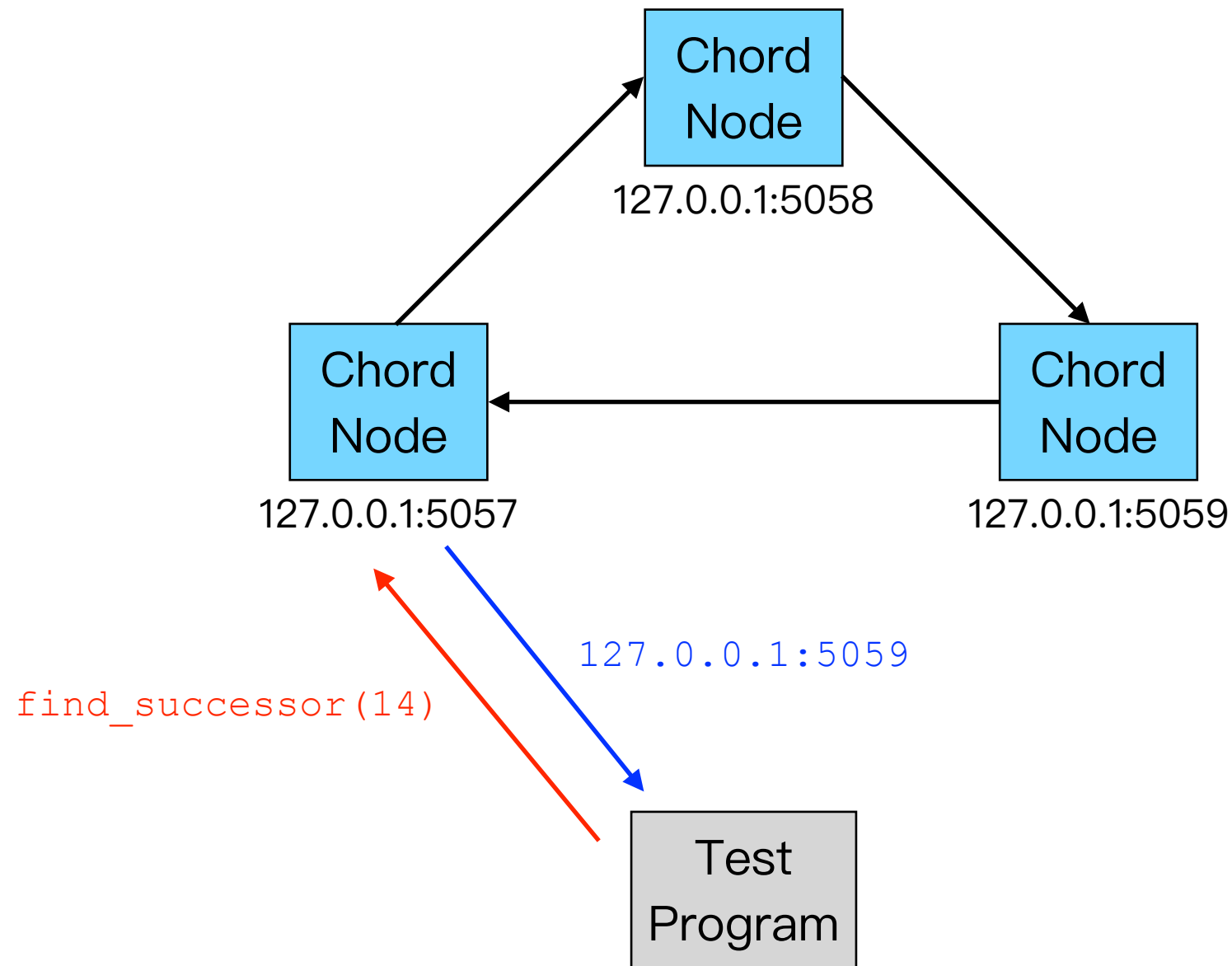
# Schedule

Week 4 (3/16)	Part 1 Announced
Week 5 (3/23)	
Week 6 (3/30)	Part 1 Due, Part 2 Announced
Week 7 (4/6)	Part 3 Announced (TBD)
Week 8 (4/13)	
Week 9 (4/20)	Midterm Exam
Week 10 (4/27)	Part 2 & Part 3 Due (4/30) (TBD)
Week 11 (5/4)	Demo (5/1 – 5/3)

# Part 1: DHT Layer Implementation

- Implement Chord node
- Test your implementation locally

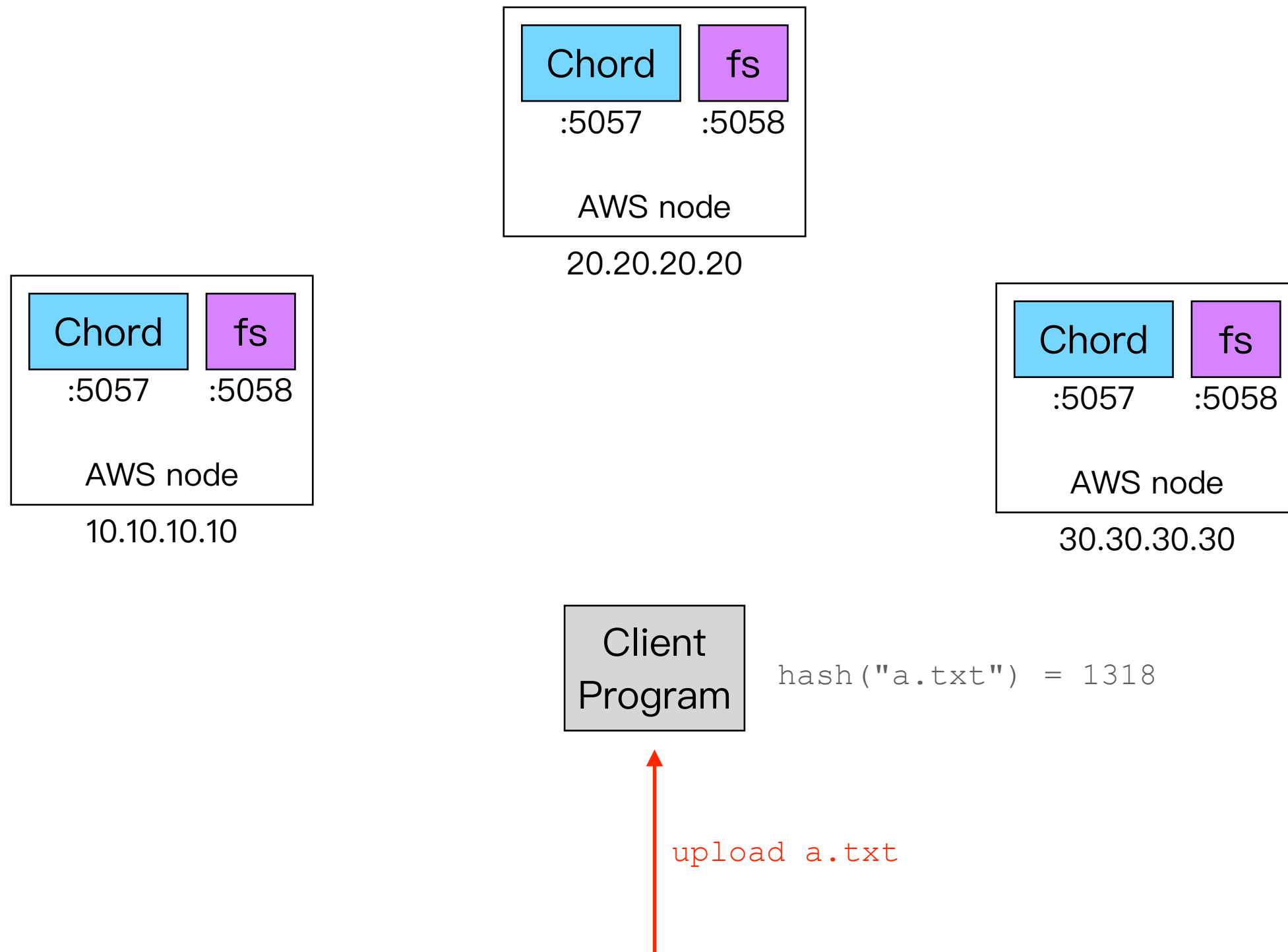
# Part 1: DHT Layer Implementation



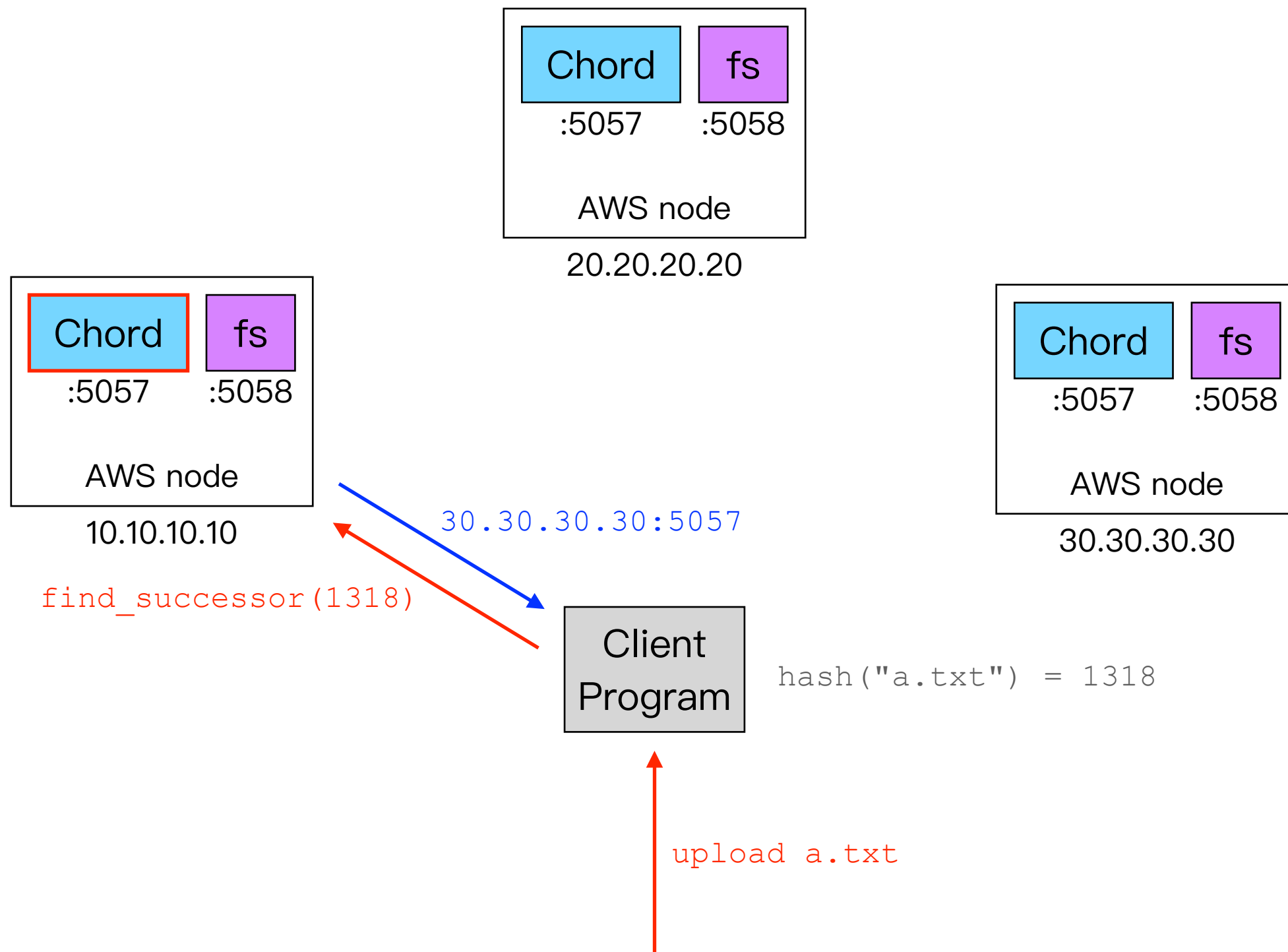
# Part 2: Automatic scaling on AWS Cloud

- Deploy Chord system and file system on AWS cloud nodes
- Upload or download files to/from an AWS cloud node
- Automatically scale the system based on the amount of storage used
- Binary will be provided for those who have not completed part 1

# Part 2: Automatic scaling on AWS Cloud

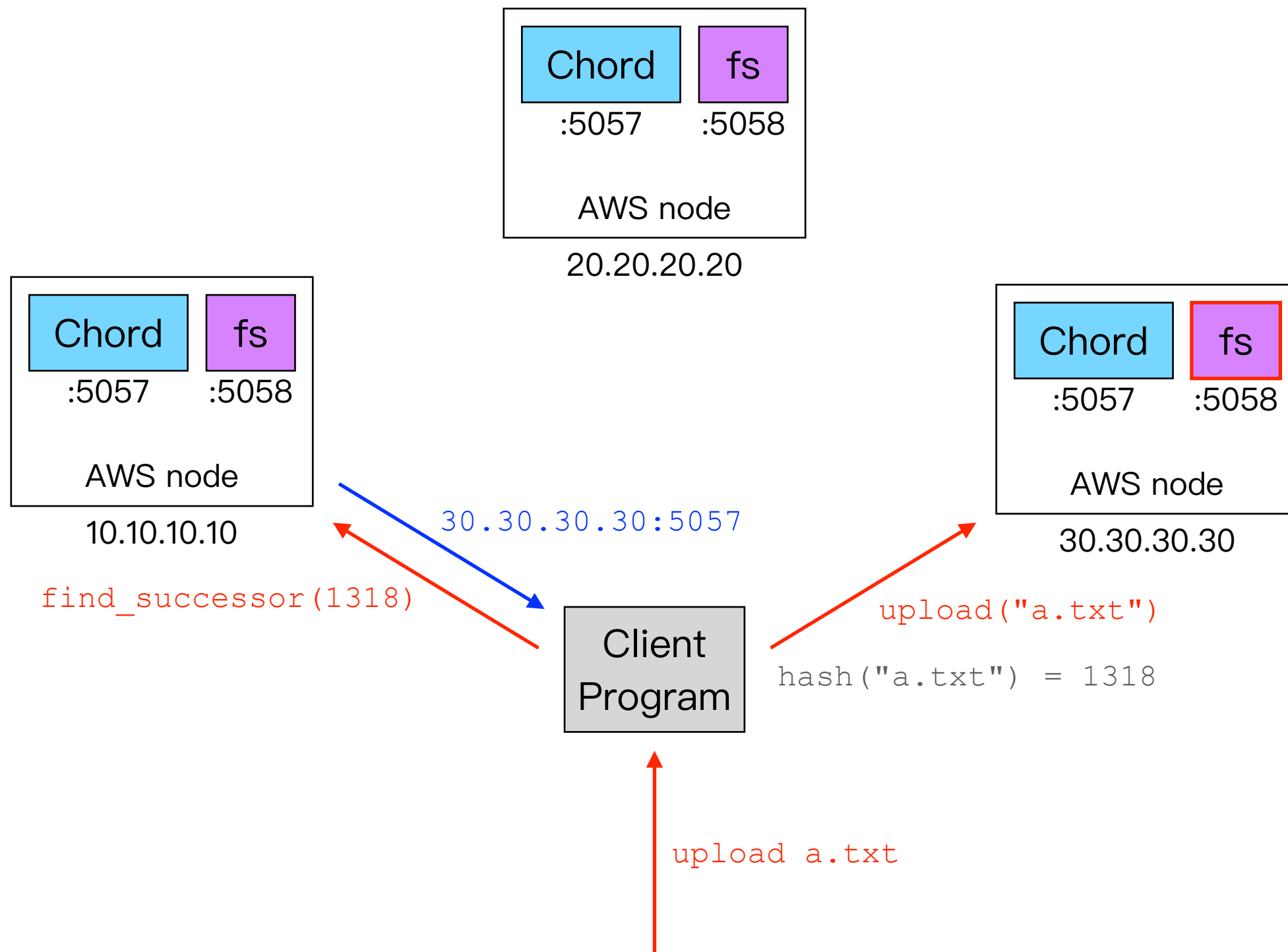


# Part 2: Automatic scaling on AWS Cloud

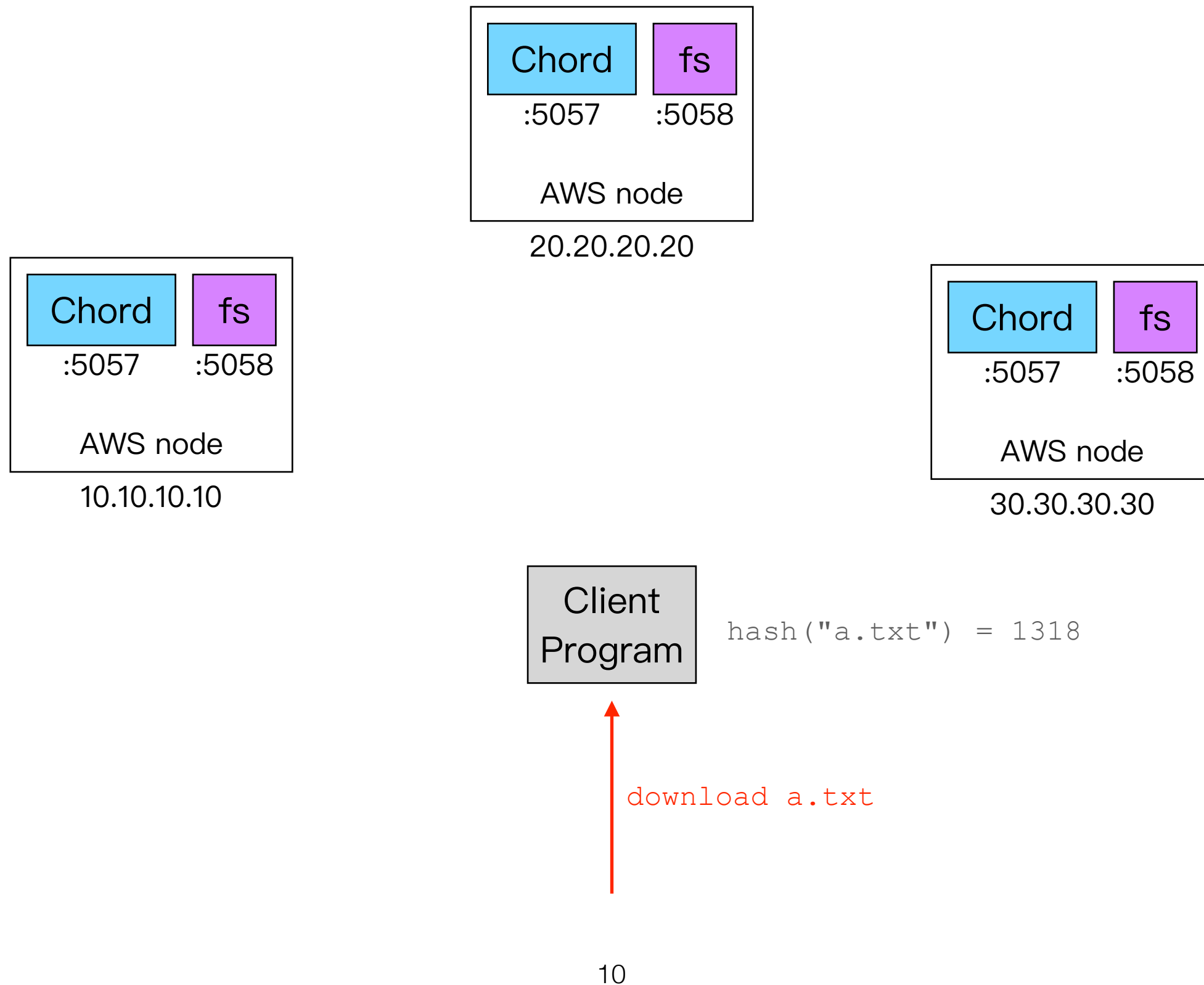




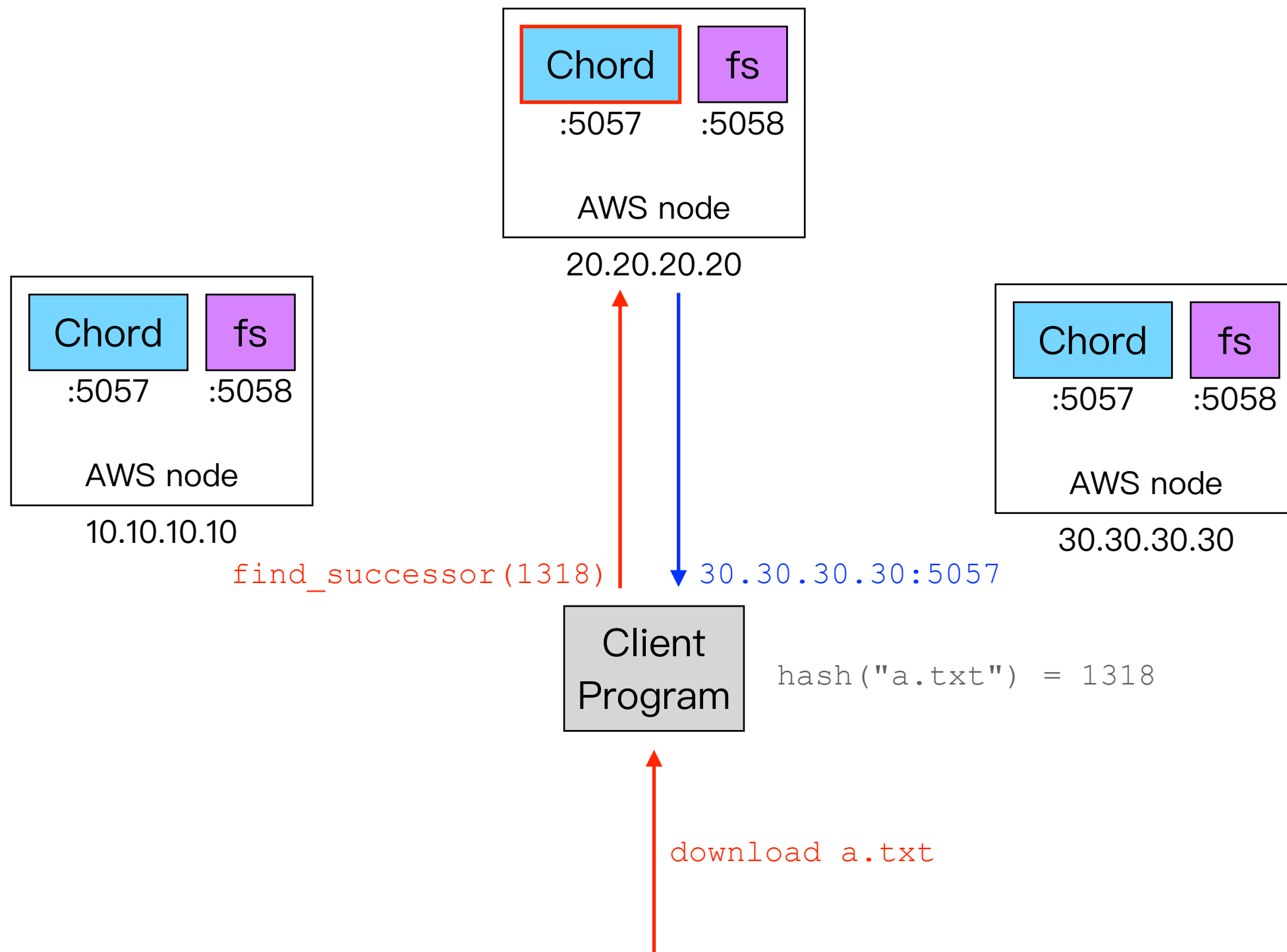
# Part 2: Automatic scaling on AWS Cloud



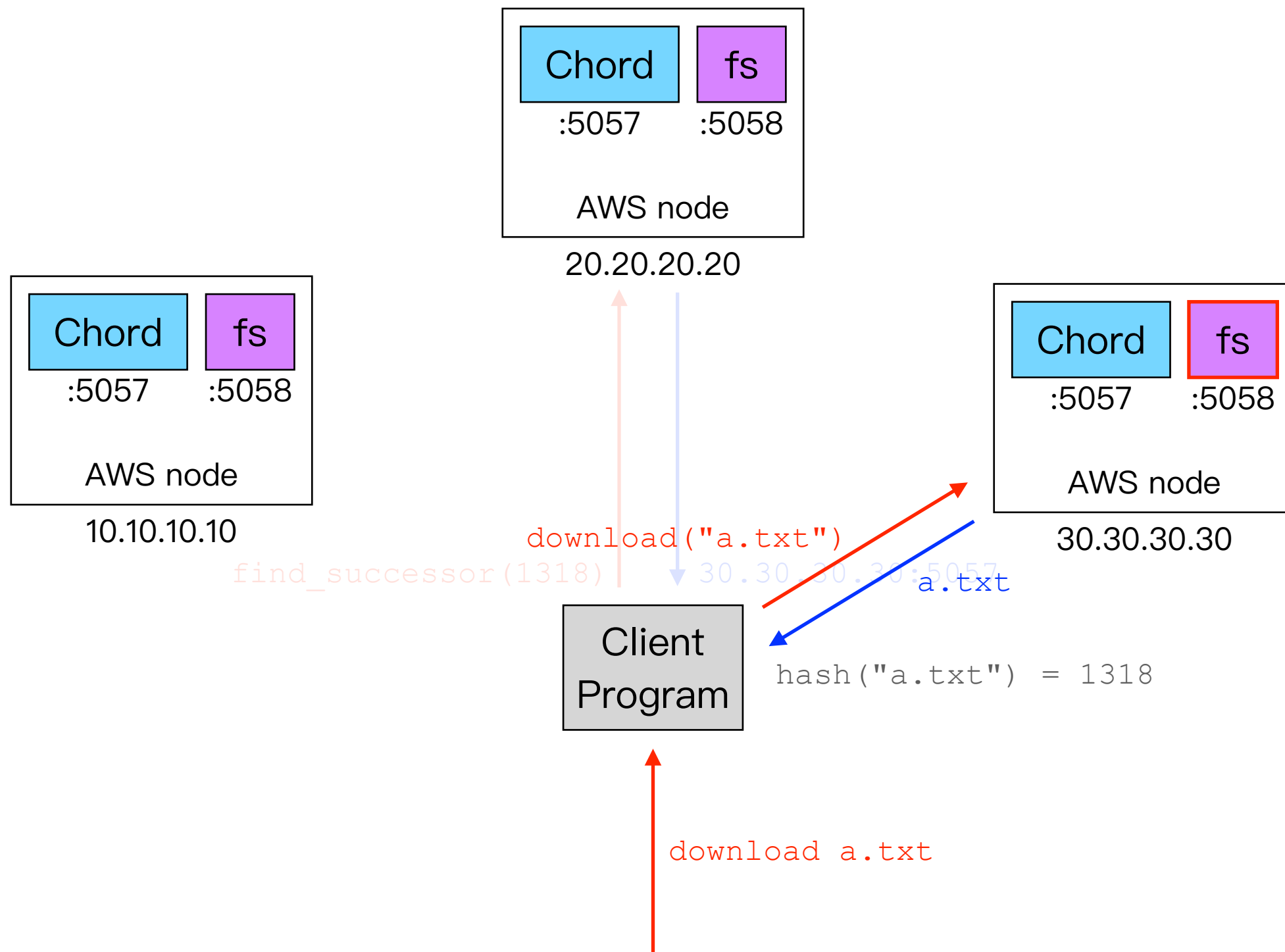
# Part 2: Automatic scaling on AWS Cloud



# Part 2: Automatic scaling on AWS Cloud



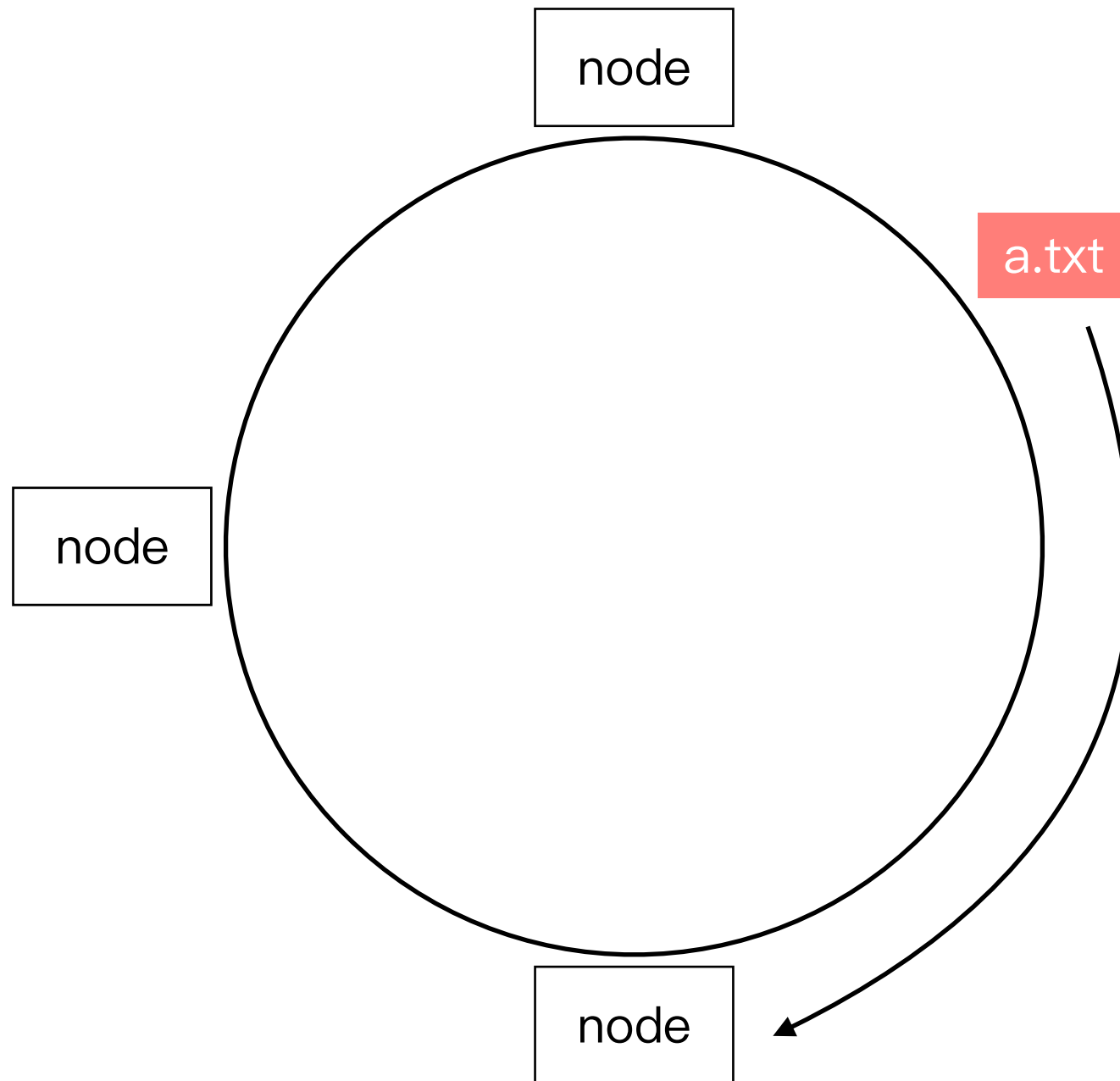
# Part 2: Automatic scaling on AWS Cloud



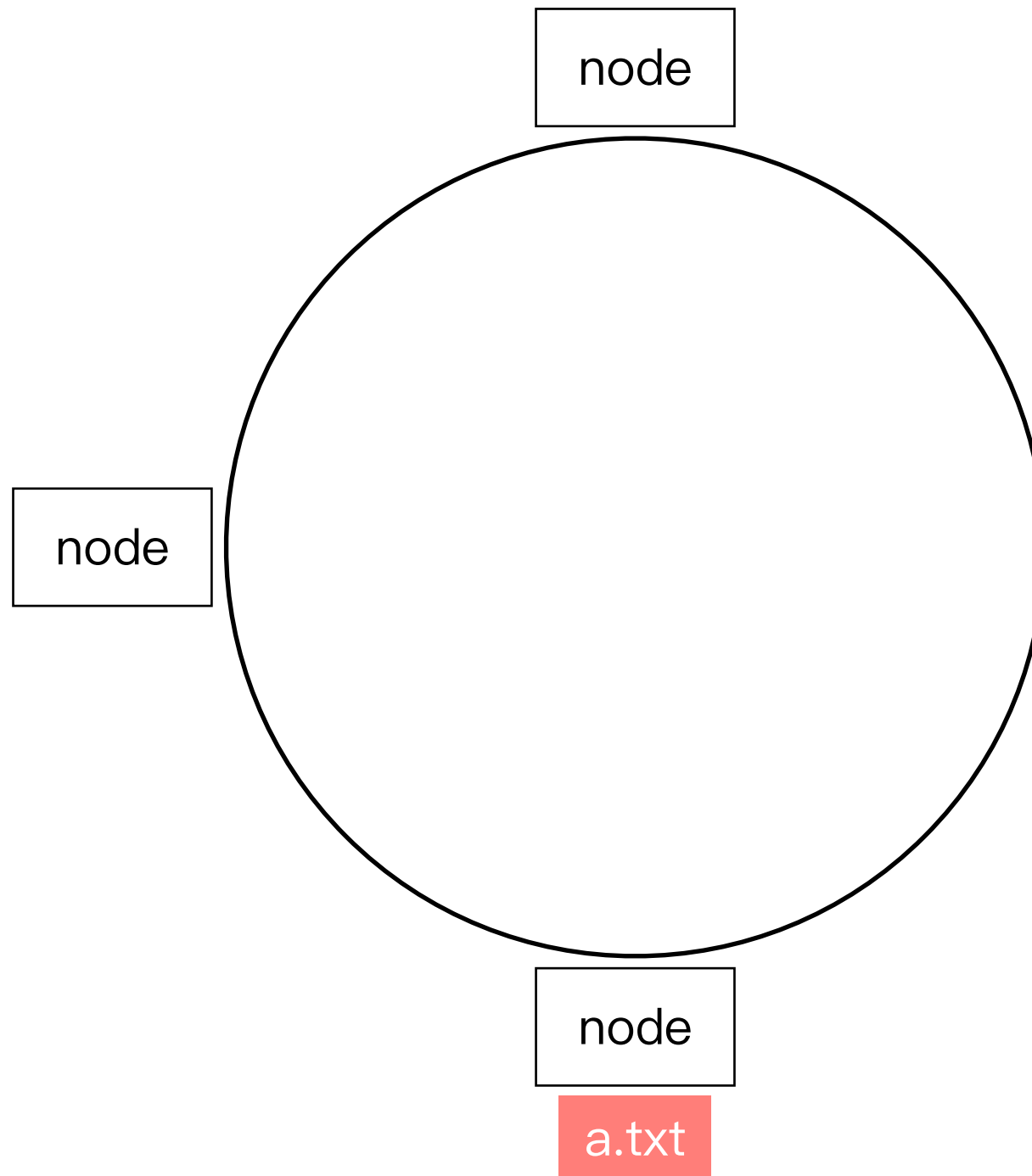
# Part 3: Distributed File System Implementation

- Migrate the data when new nodes are added
- Balance the load on each node by dividing files into smaller file chunks
- Implement file replication to ensure fault tolerance

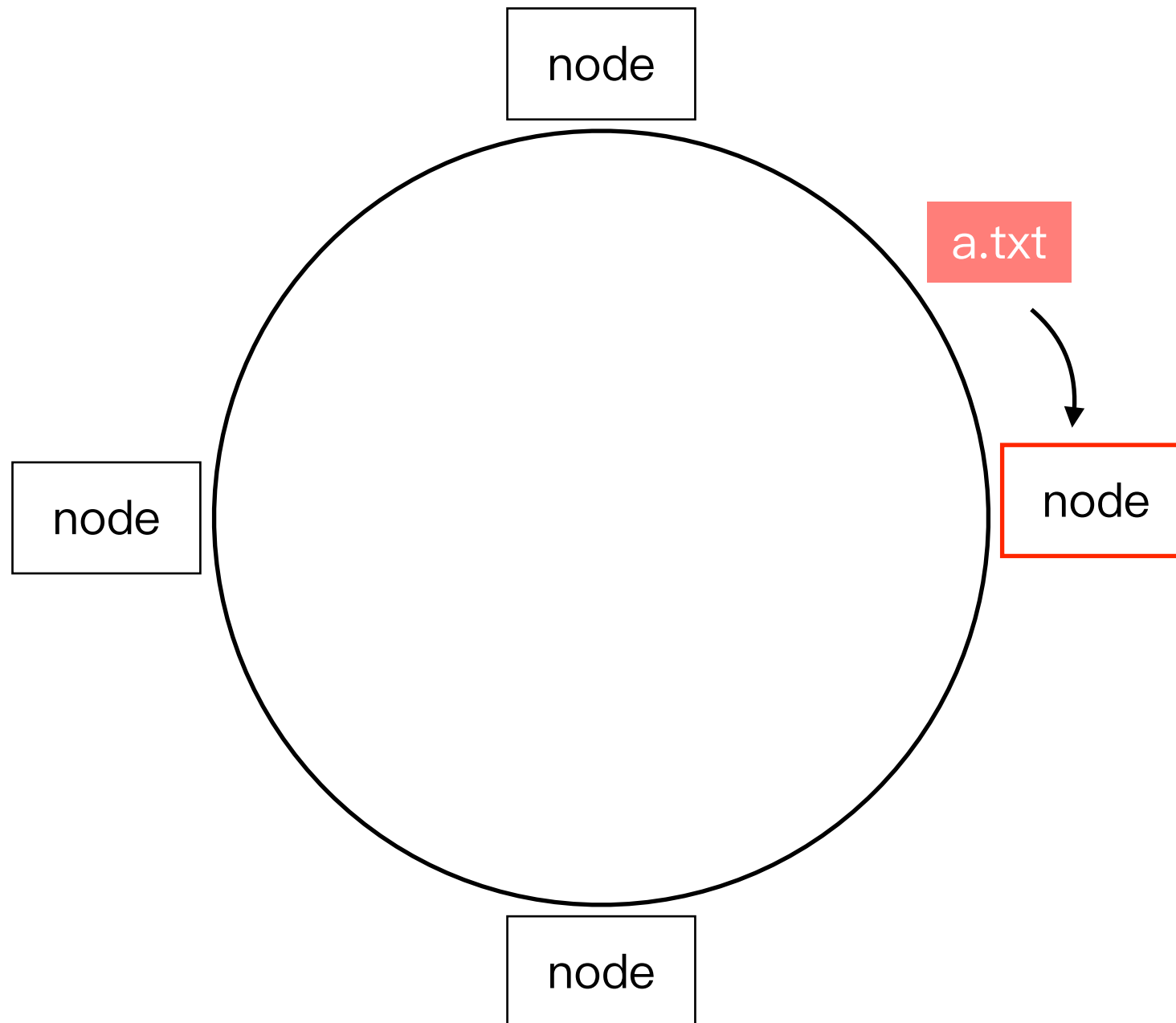
# Correctness – Data Migration



# Correctness – Data Migration

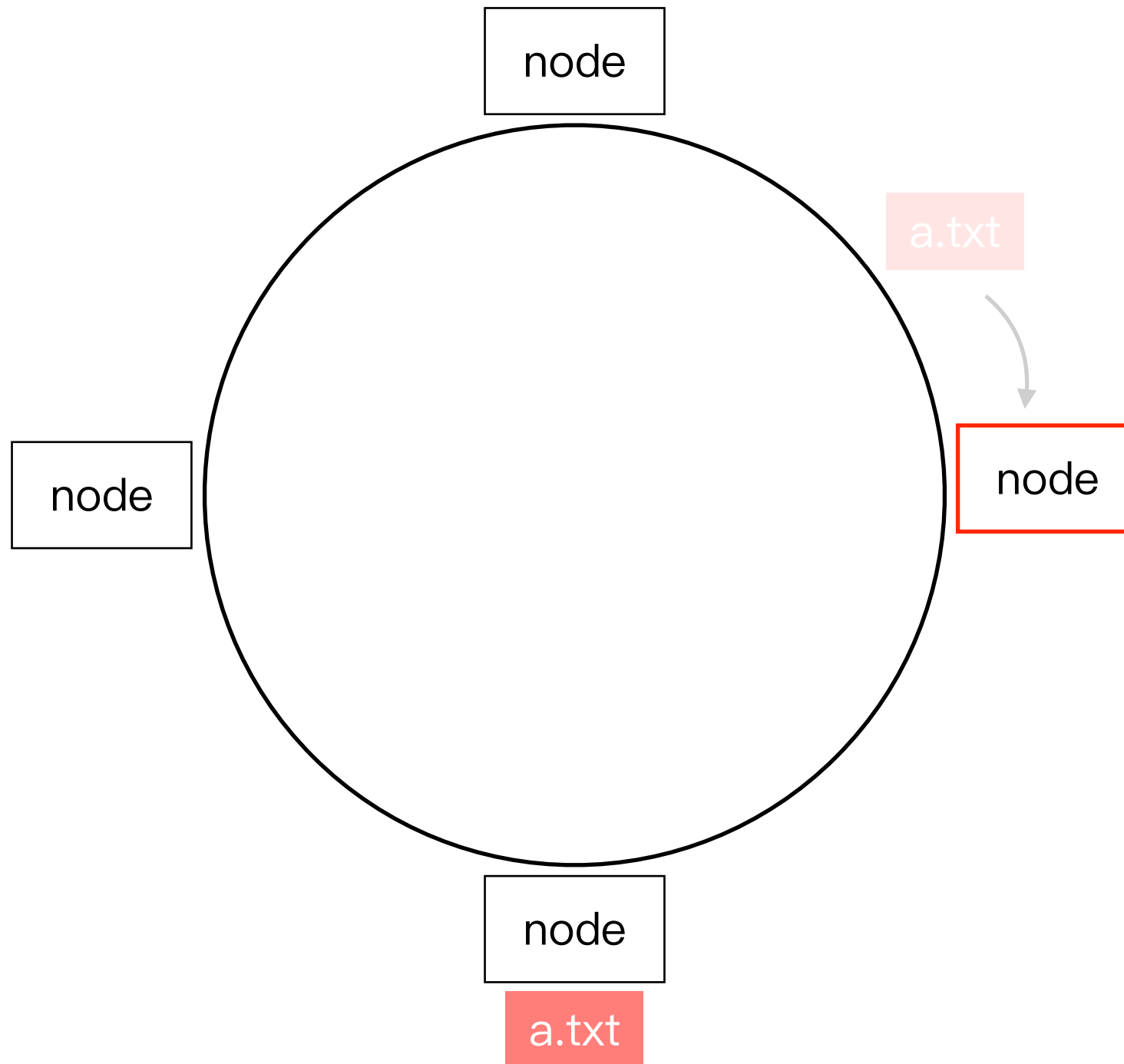


# Correctness – Data Migration

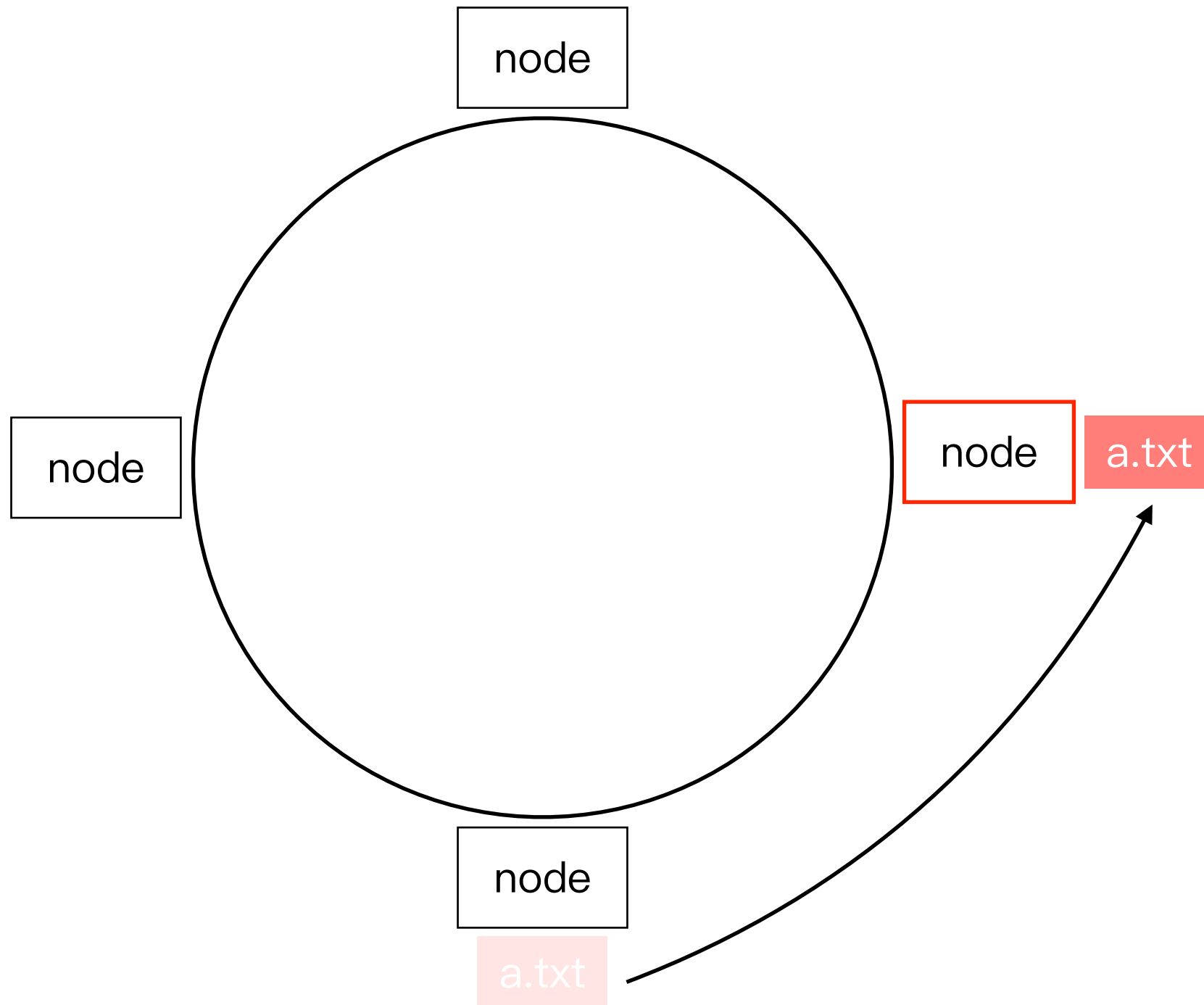




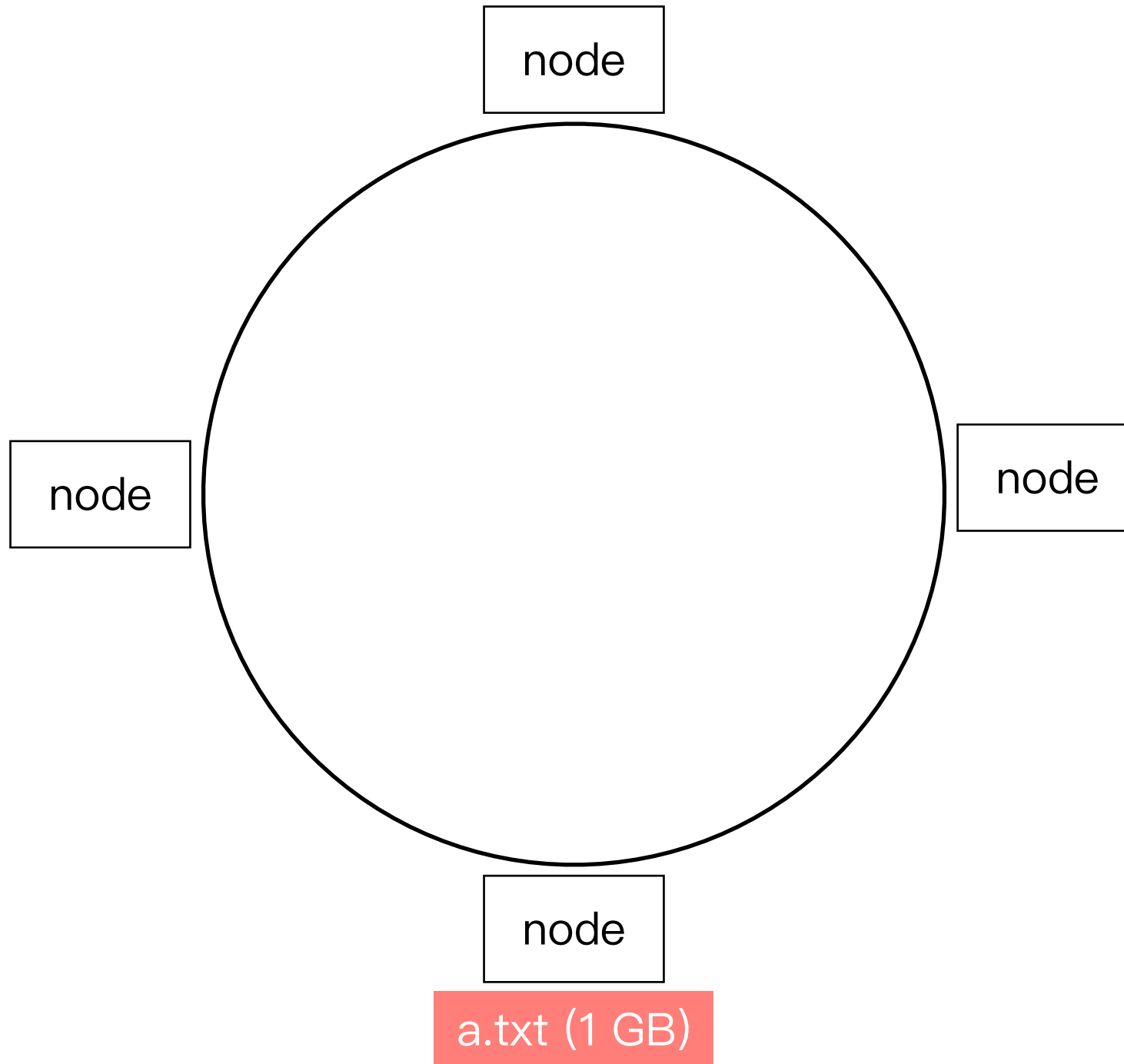
# Correctness – Data Migration



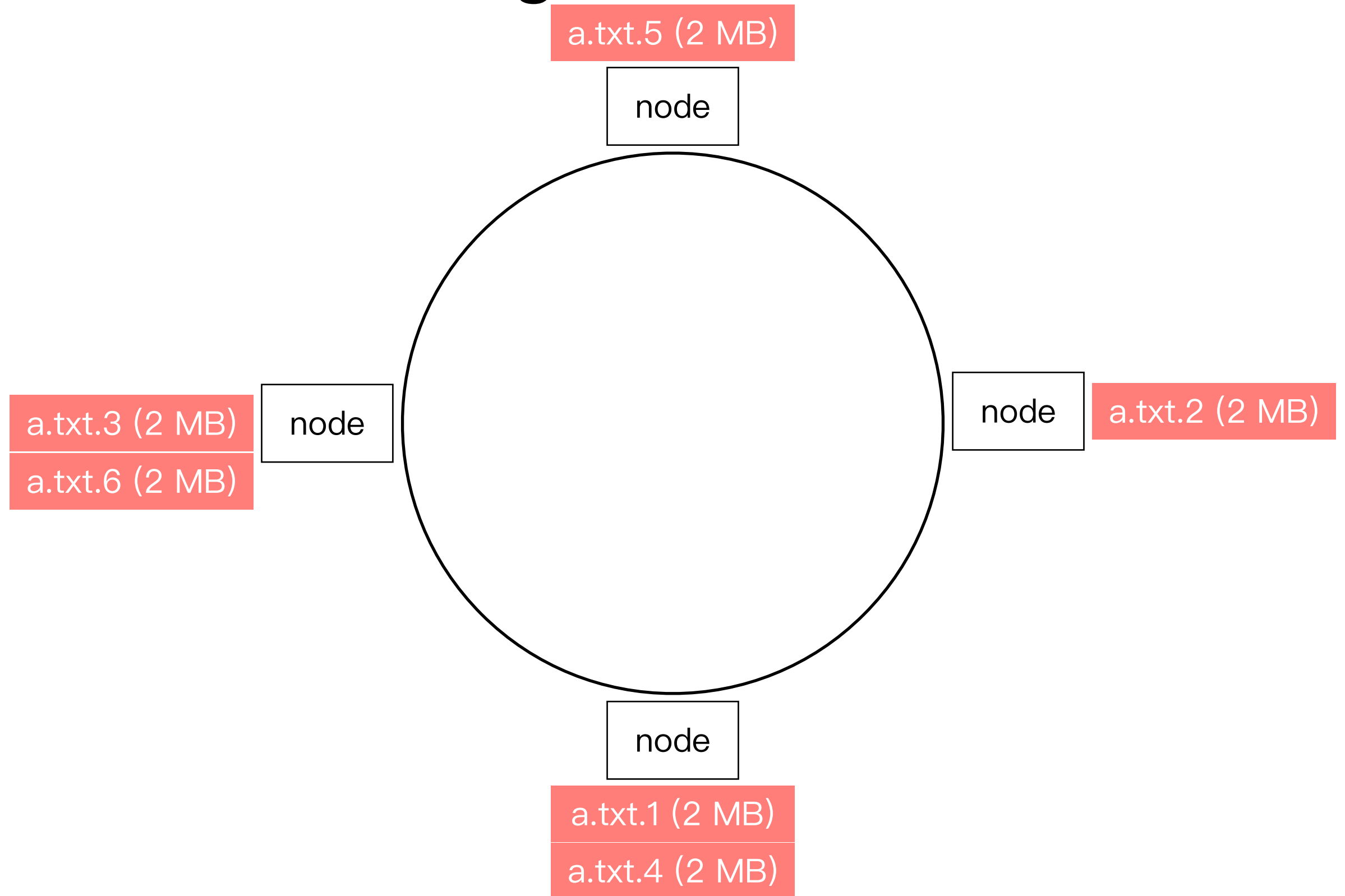
# Correctness – Data Migration



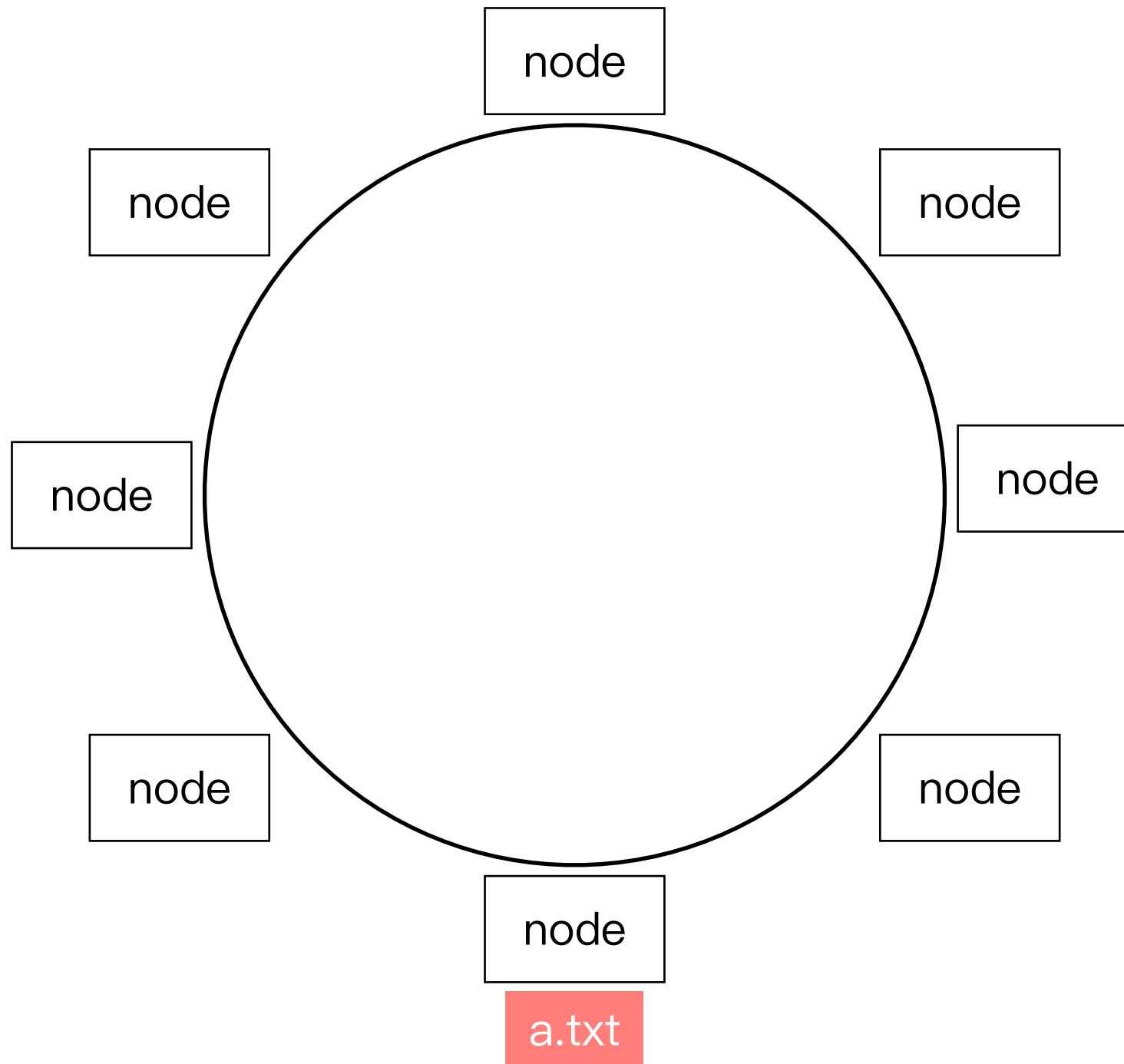
# Load Balancing – File Chunks



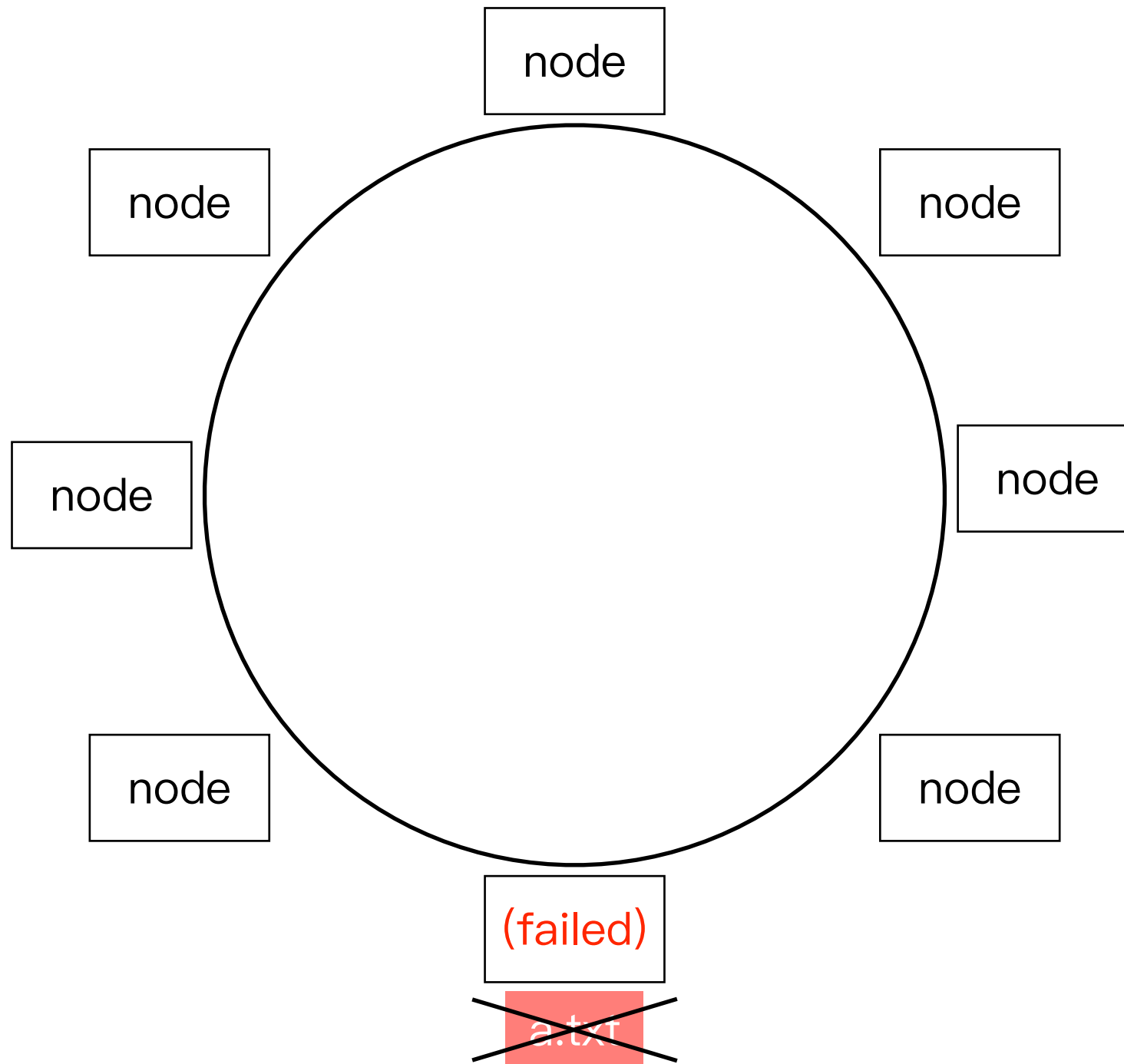
# Load Balancing – File Chunks



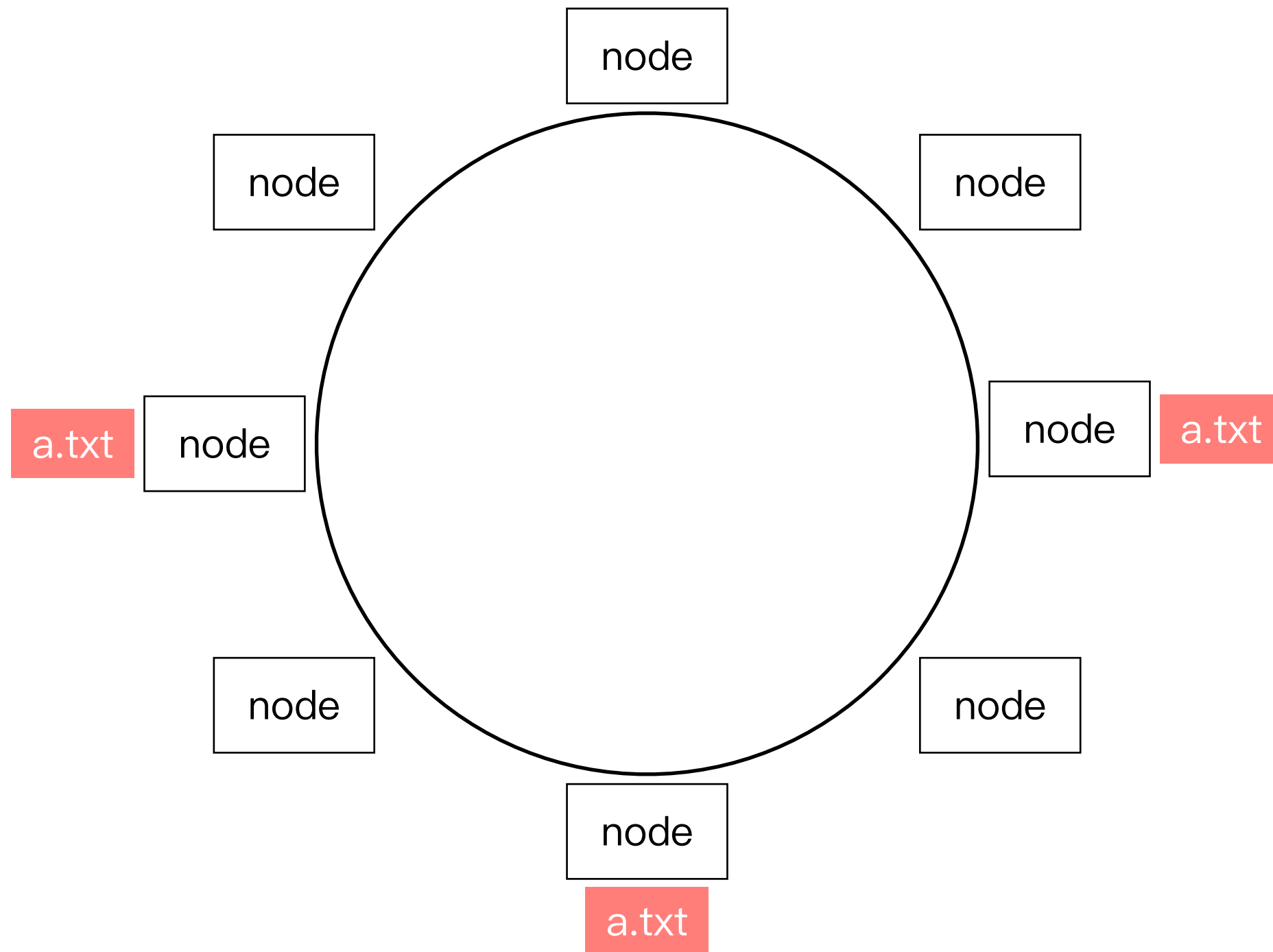
# Fault Tolerance – Replication



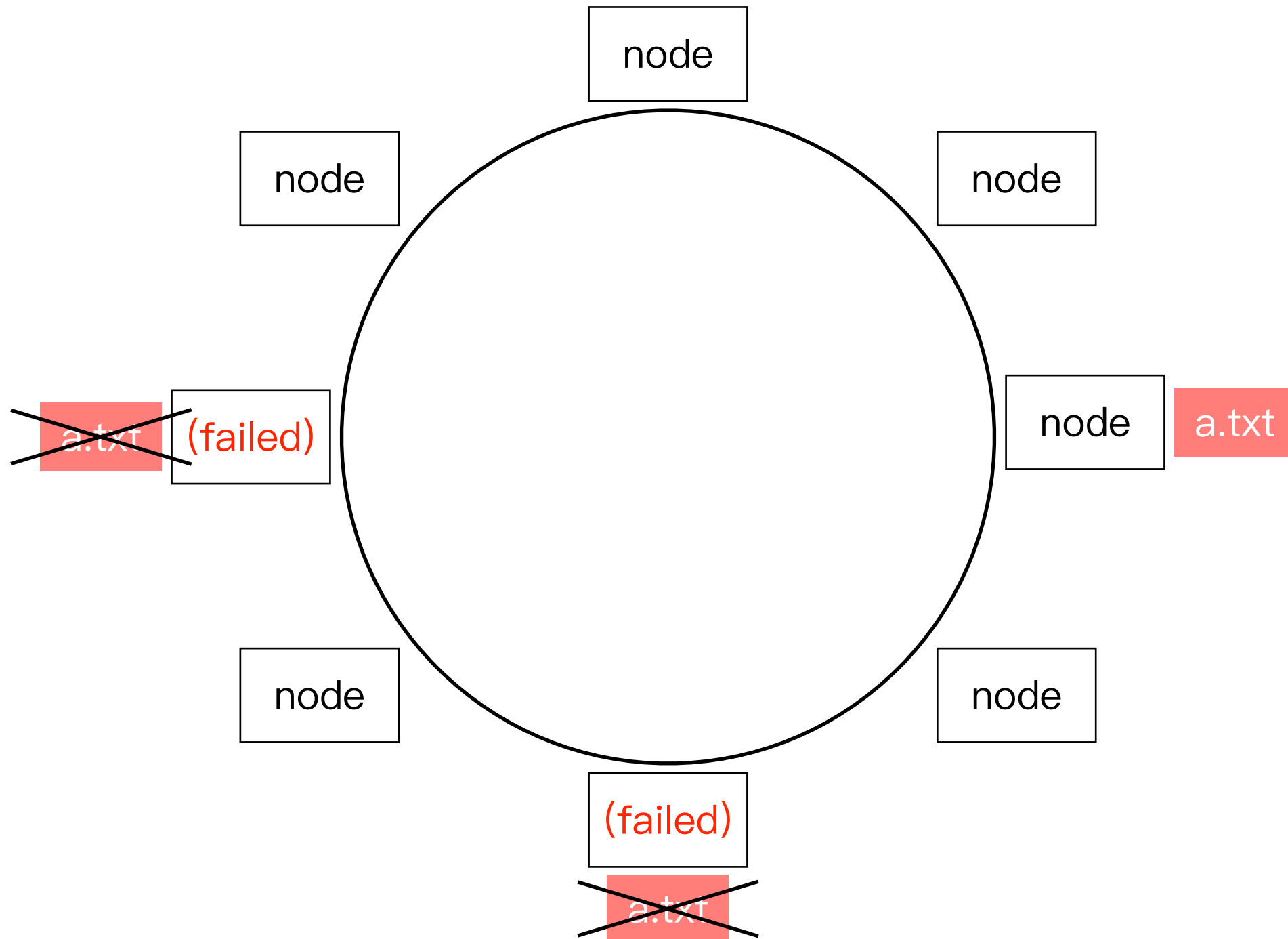
# Fault Tolerance – Replication



# Fault Tolerance – Replication

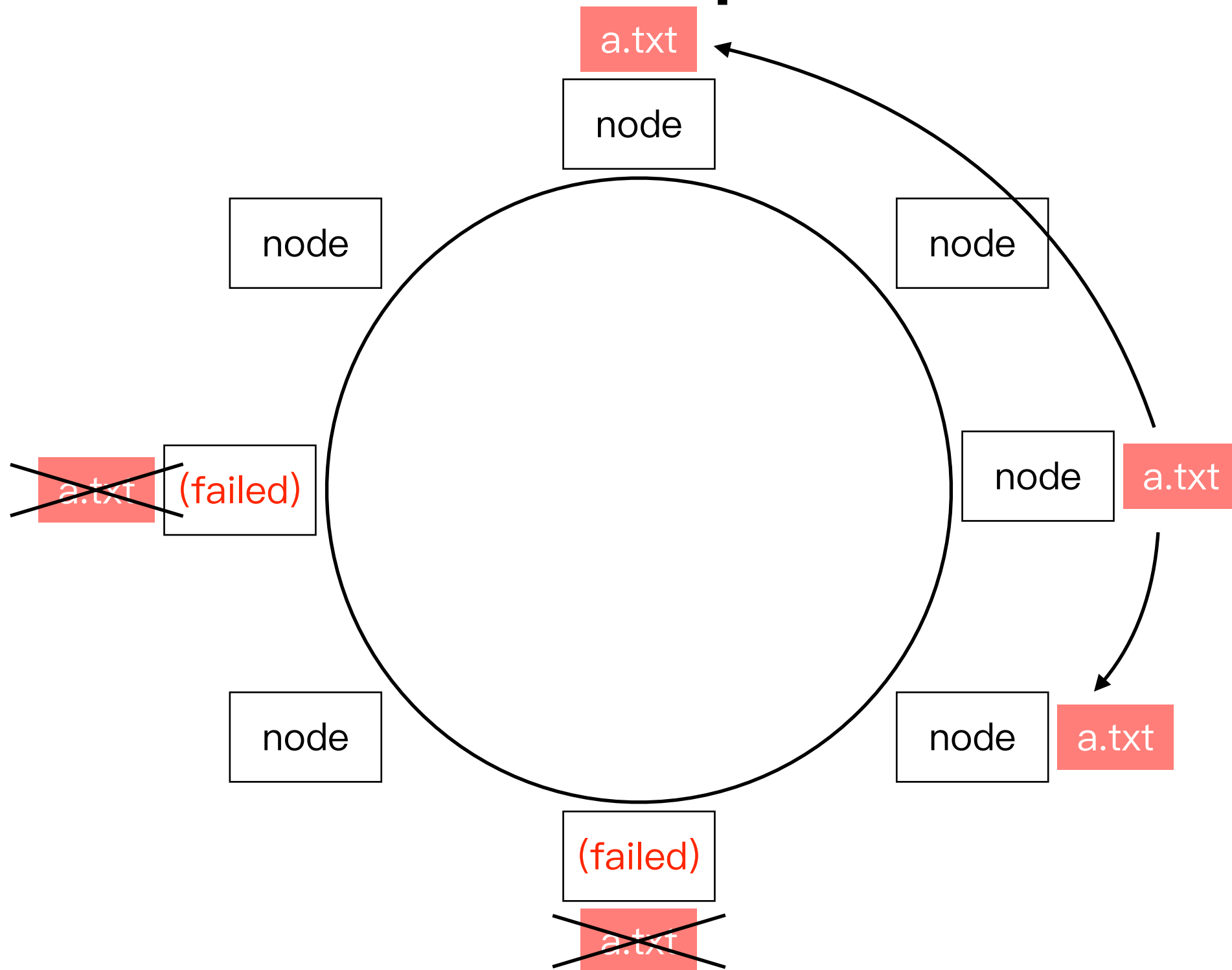


# Fault Tolerance – Replication





# Fault Tolerance – Replication



# Grading Policy

- Part 1: 10%
  - Correctness: 4%
  - Message Complexity: 3%
  - Fault Tolerance: 3%
- Part 2: 6%
- Part 3: 9%
  - Data Migration: 3%
  - File Chunks: 3%
  - Replication: 3%