

# A Fault-Tolerant Transparent Data Sharing Service for the Grid

**Louis Rilling**, Christine Morin  
IRISA / PARIS research group

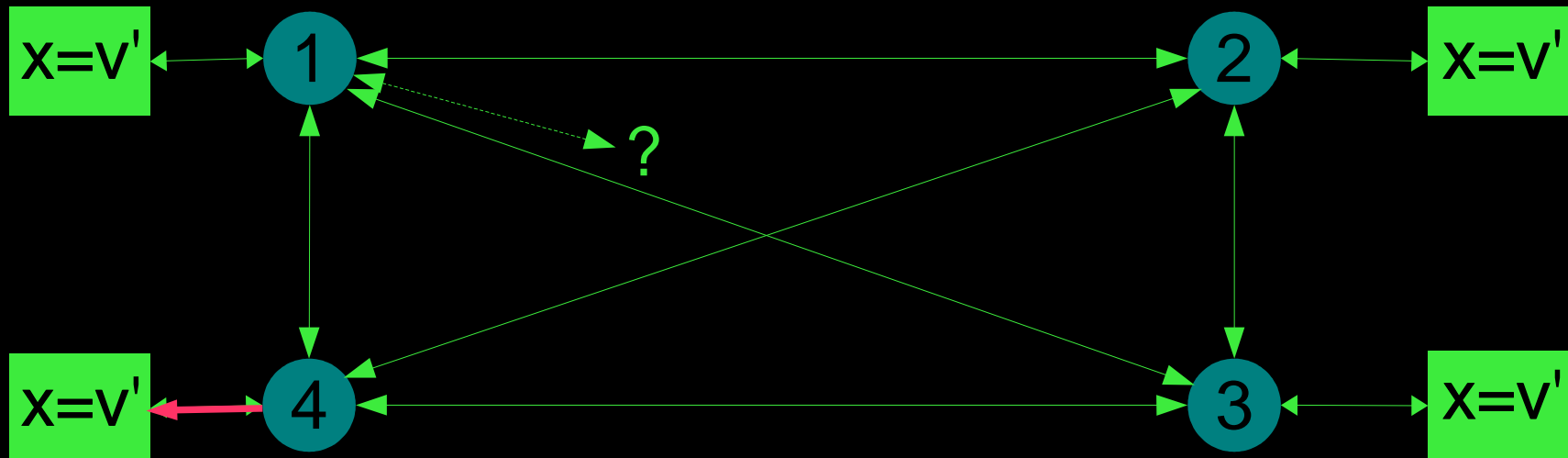


# Objectives

- Why data sharing should be transparent and fault-tolerant?
- How to obtain transparency and tolerance to multiple failures?

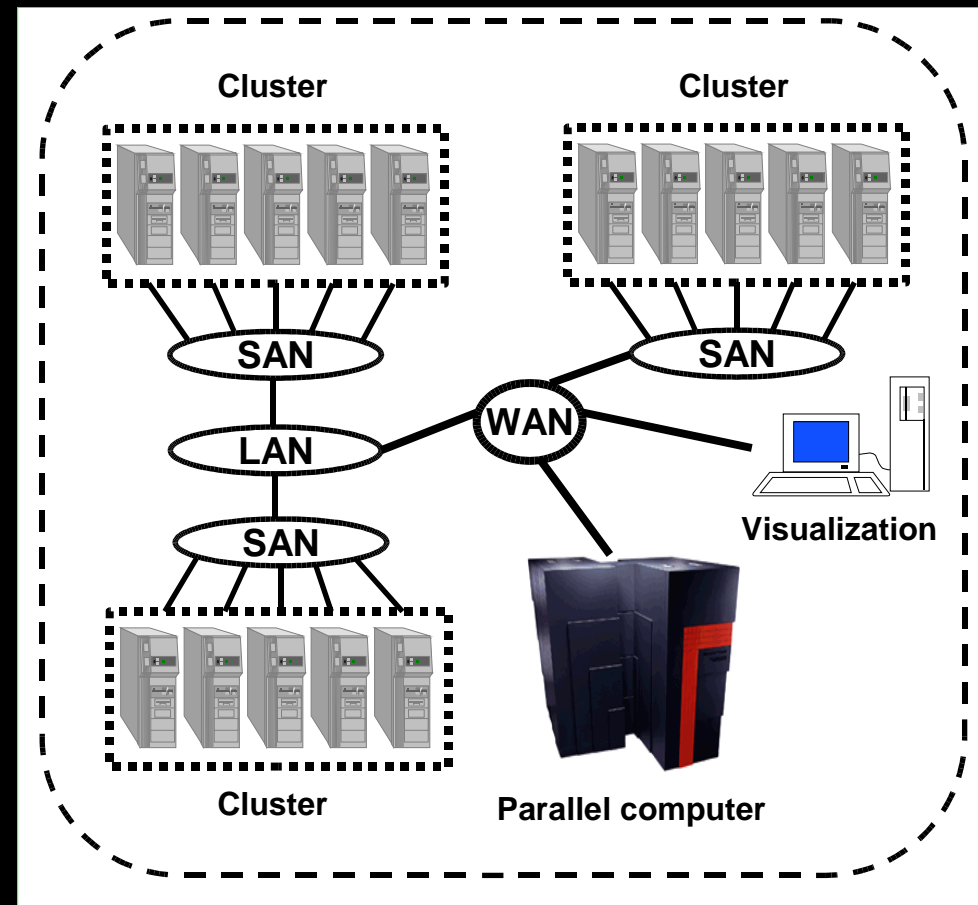
# Data Sharing is Complex

- Applications of distributed computing
- Localize
- Cache and replicate
- Keep replica consistent

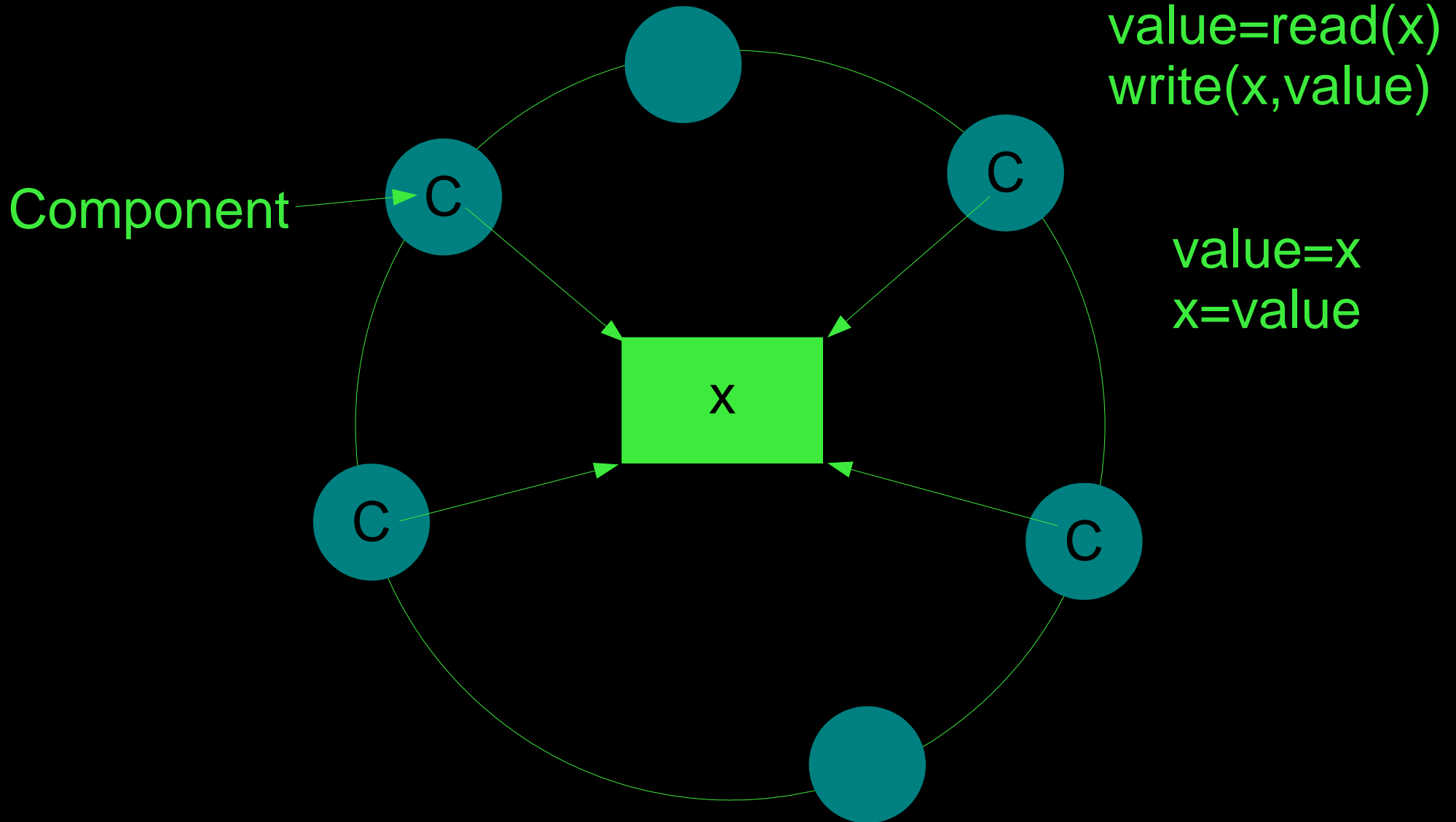


# In a Grid, this is Worse

- Sites independent numerous
- Dynamic configuration
  - Reconfigurations / Failures
  - Many and simultaneously



# Data Sharing with Atomic Consistency: Logical View



# Single System Image Approach

- Hide the distributed aspect of data
- Access data using location-independent names
- Atomic consistency protocol to locate, cache, and keep copies consistent
  - Tolerates multiple and simultaneous failures

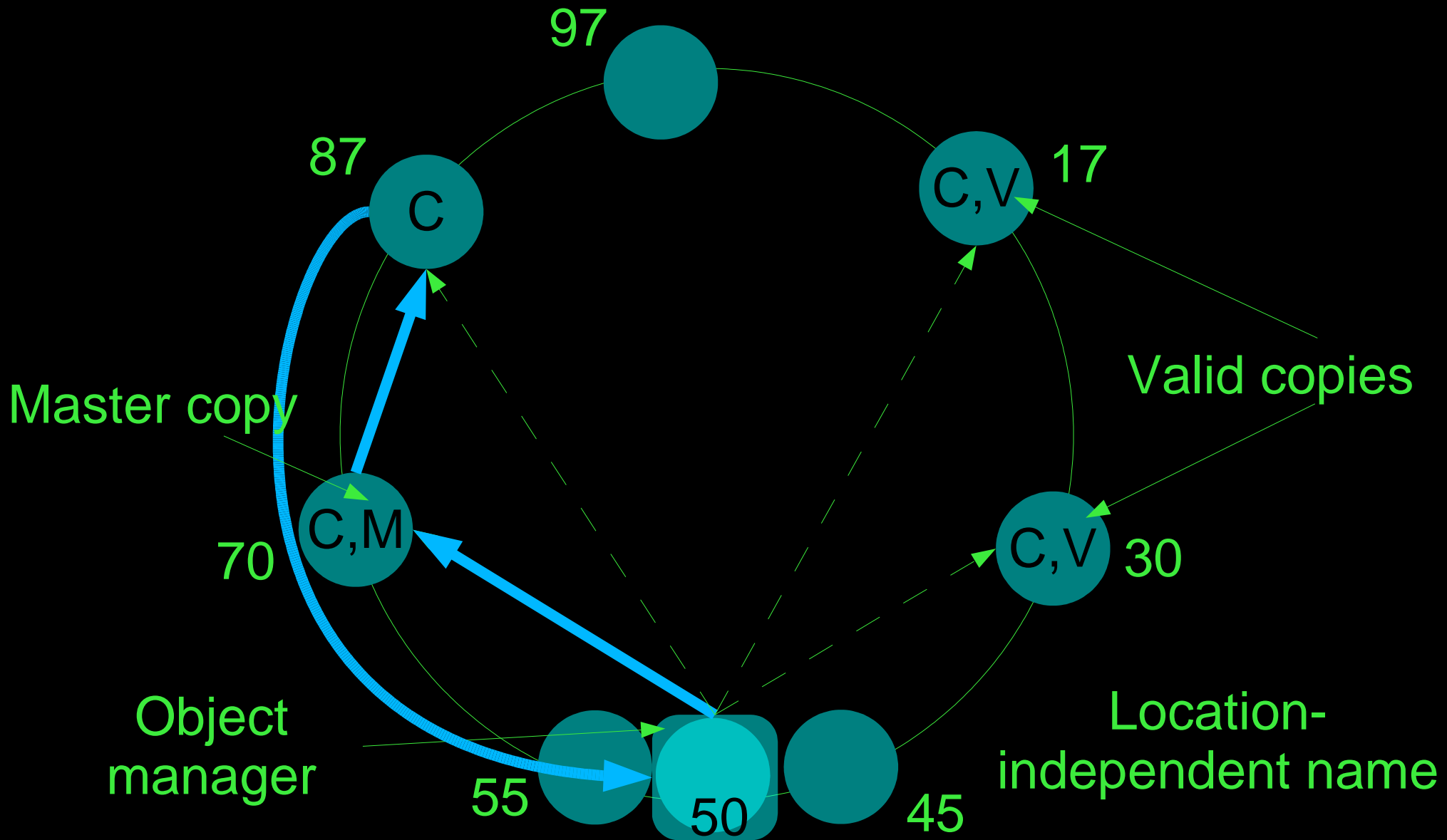
# State of the Art

- Atomic memory
  - Active replication -> write-multicast
  - High degree of fault-tolerance
  - High latencies
  - Simplify applications?
- DSM
  - Write-invalidate
  - Low degree of fault-tolerance
  - Optimized latencies
  - (Transparent) Checkpoint / restart

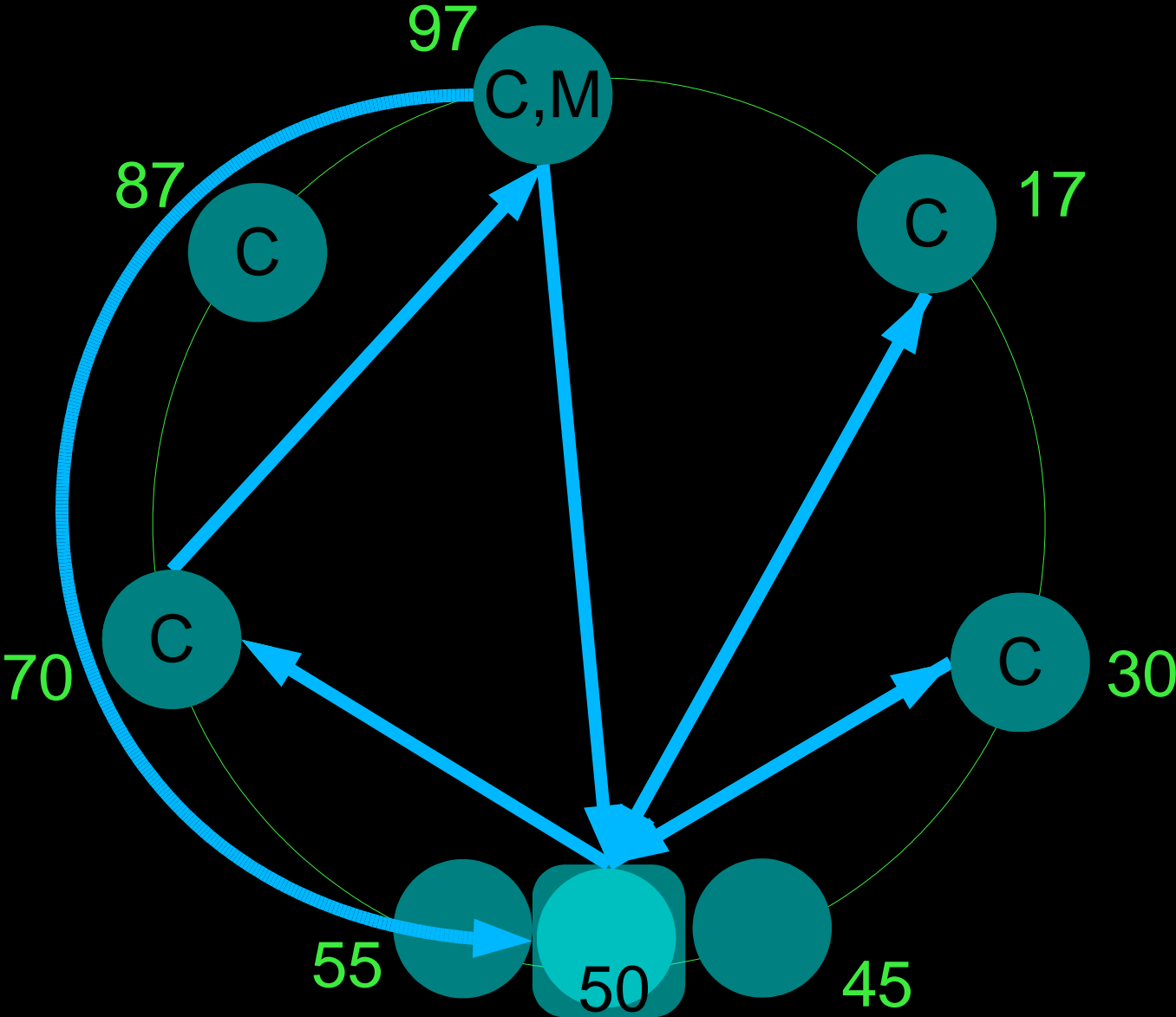
# Method

- Inspire from an existing consistency protocol
  - Write-invalidate
  - K. Li, static distributed managers
- Adapt the protocol to unreliable communications
  - Messages received in disorder, duplicated
- Tolerate multiple and simultaneous reconfigurations
  - Access managers through a DHT / P2P

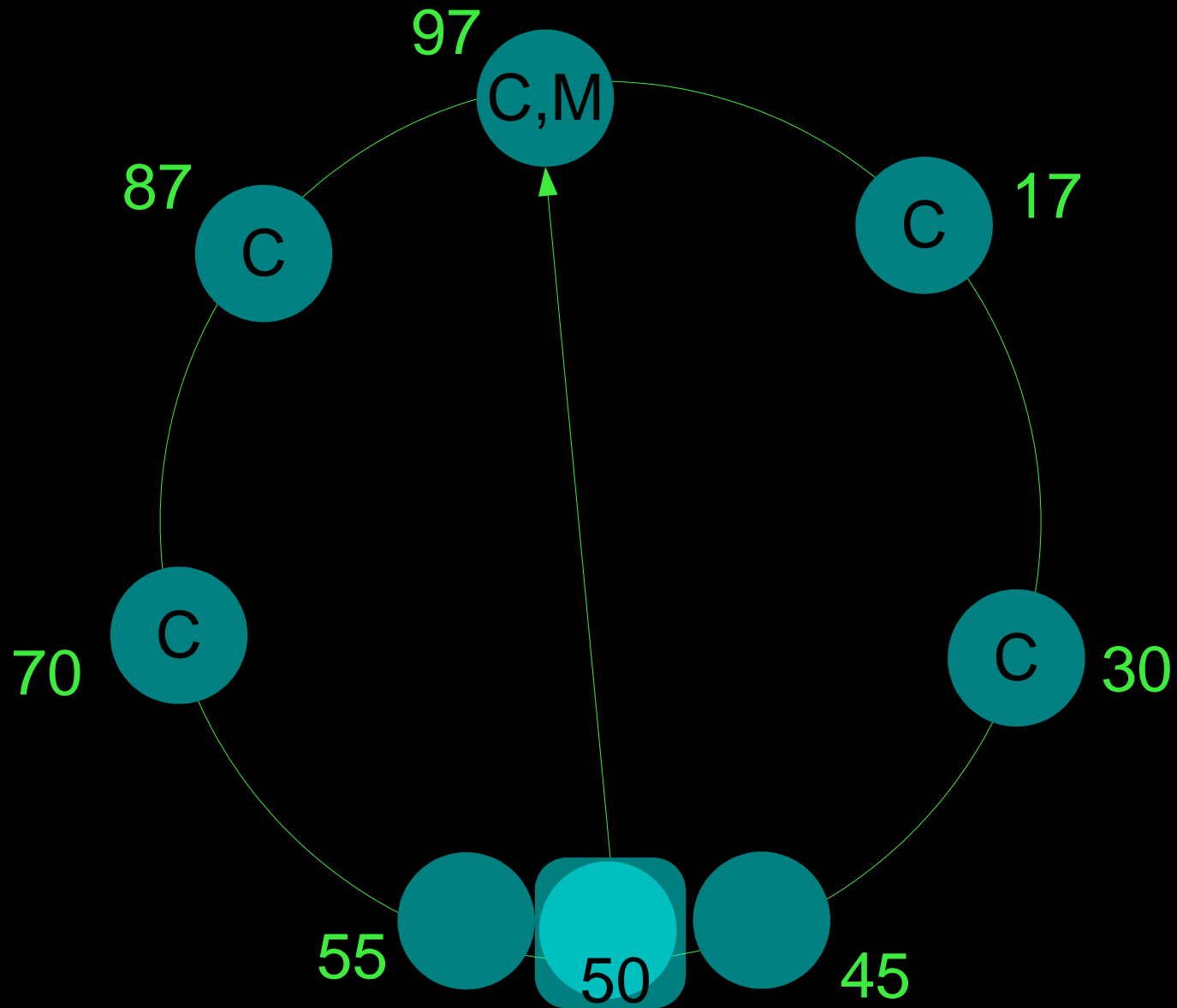
# Protocol for Atomic Consistency



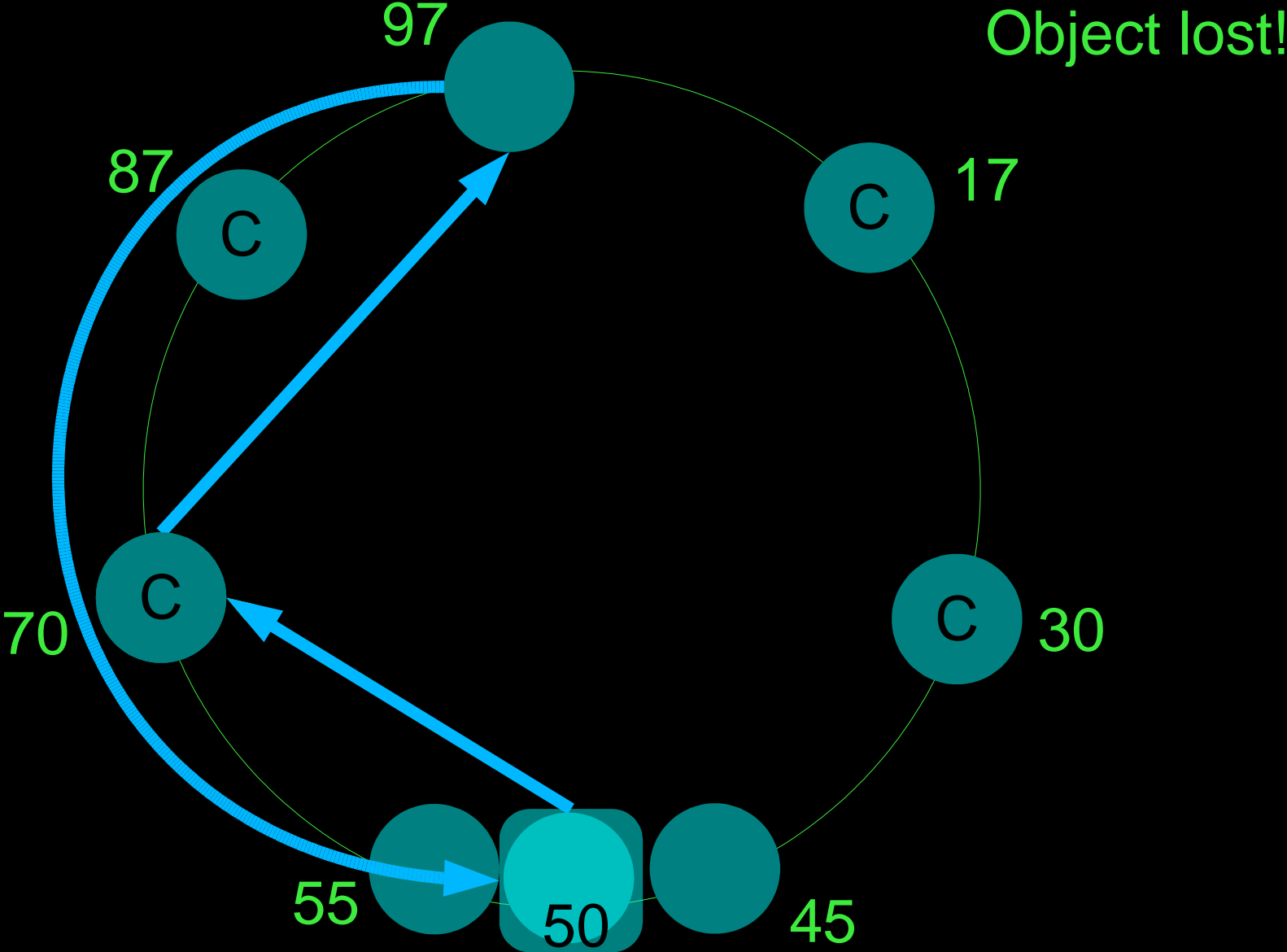
# Duplicated Messages



# Duplicated Messages



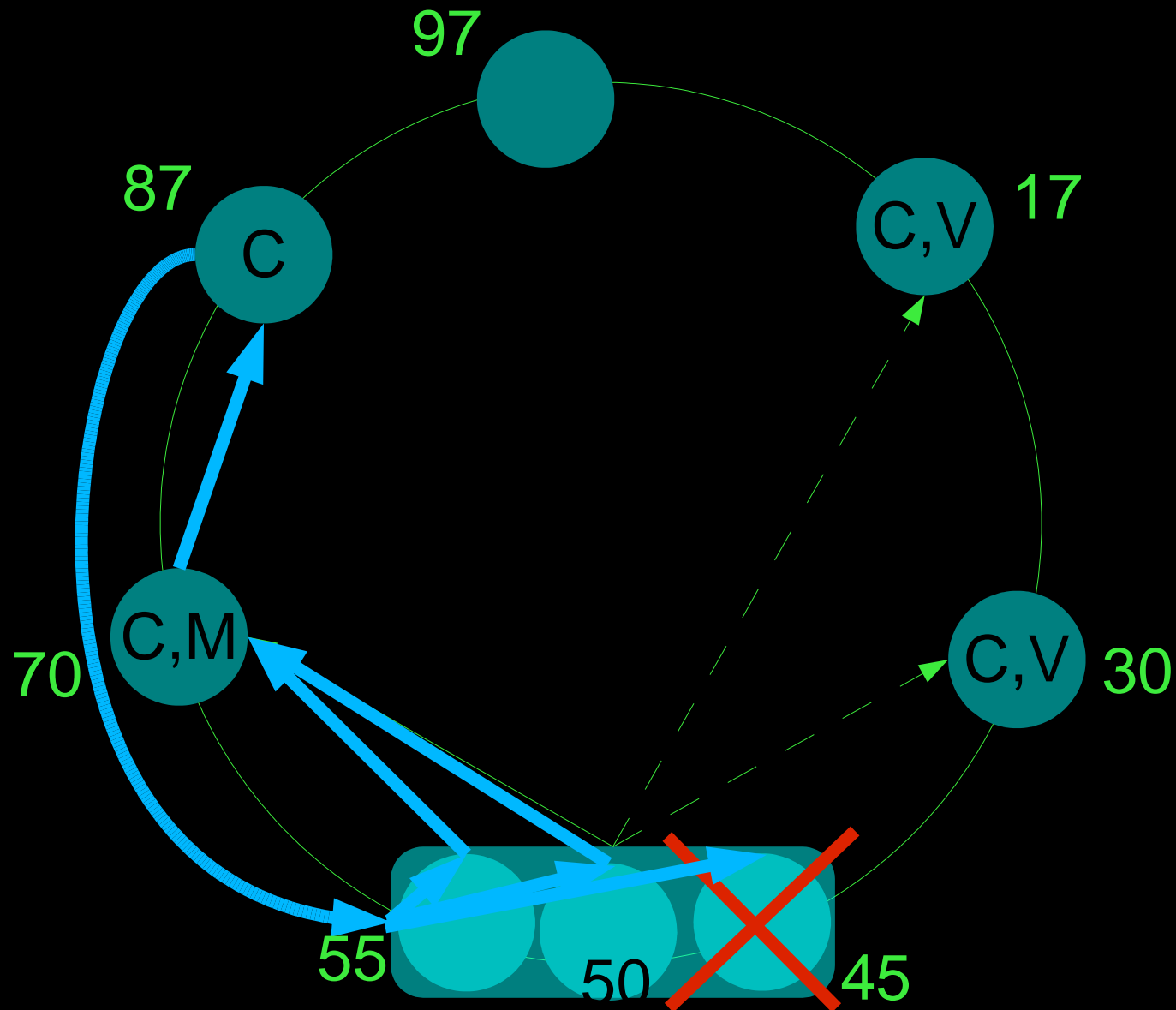
# Duplicated Messages



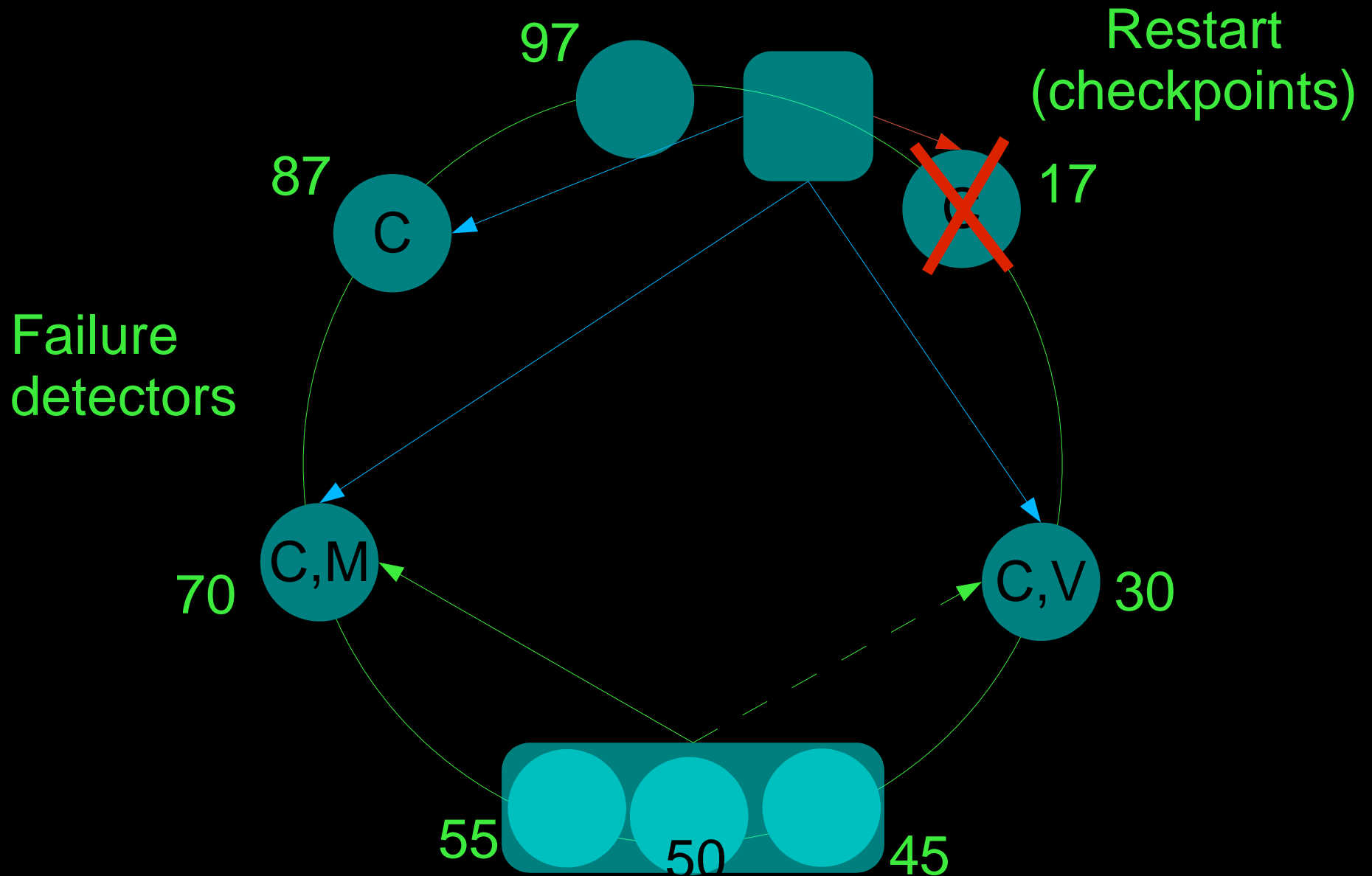
# Version Numbers

- Incremented by the object manager at each write request
- Write requests need to be confirmed
  - Confirmation request includes the new version number
  - Answer includes the new version number -> answer is fresh

# Object Manager Actively Replicated

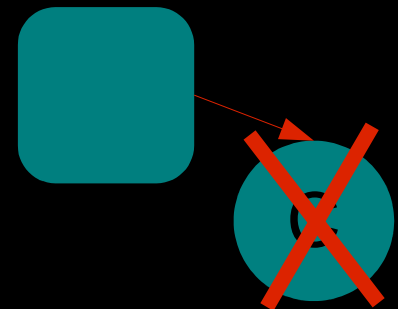
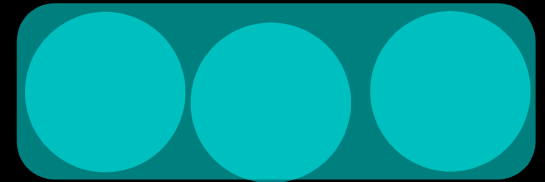


# Application Manager

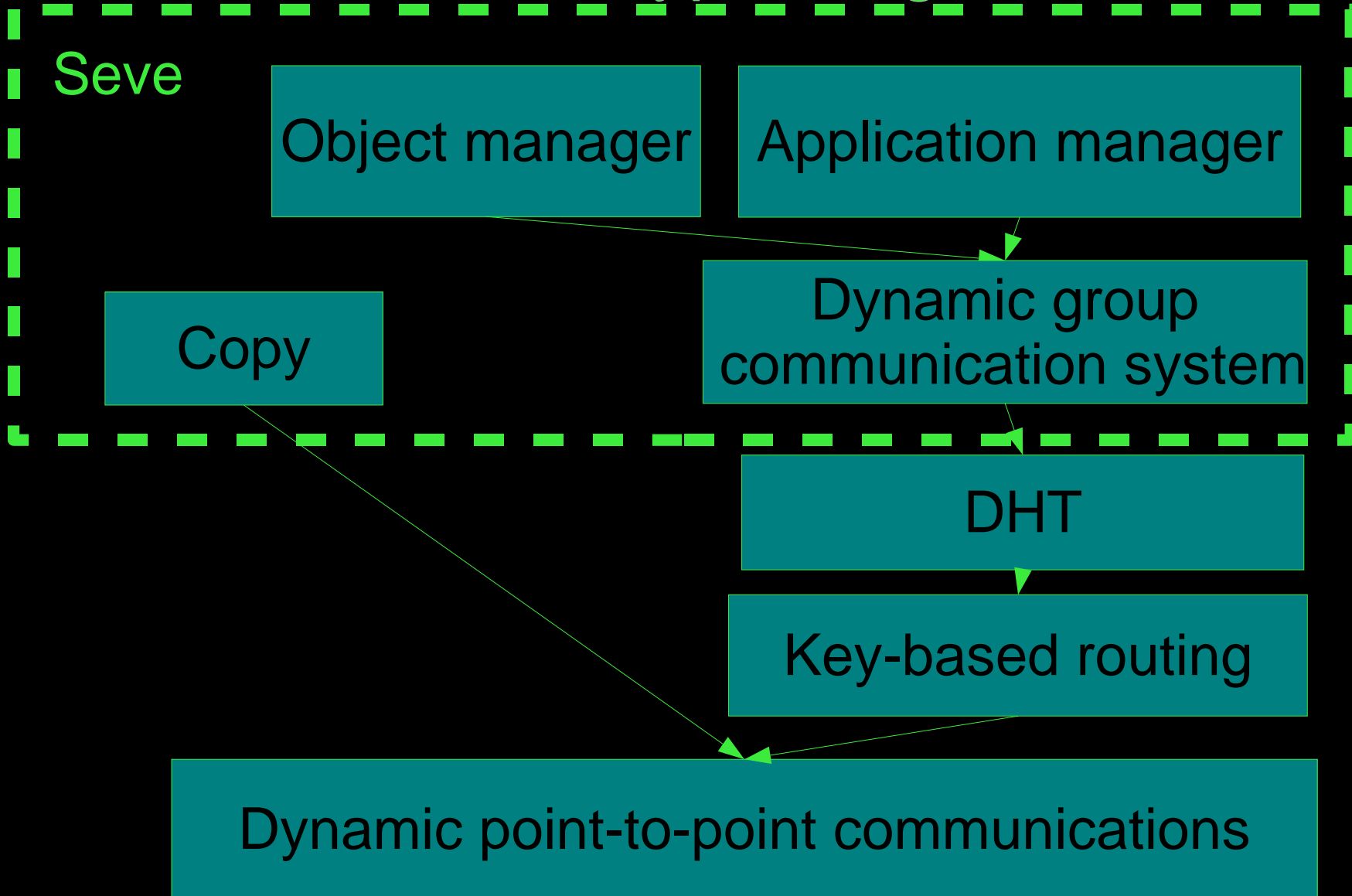


# Summary

- Version numbers
- Object manager in DHT
  - Active replication
  - Reconfigurations simplified
- Failure of an execution node -> restart (checkpoints)  
+ application manager in DHT
  - Active replication

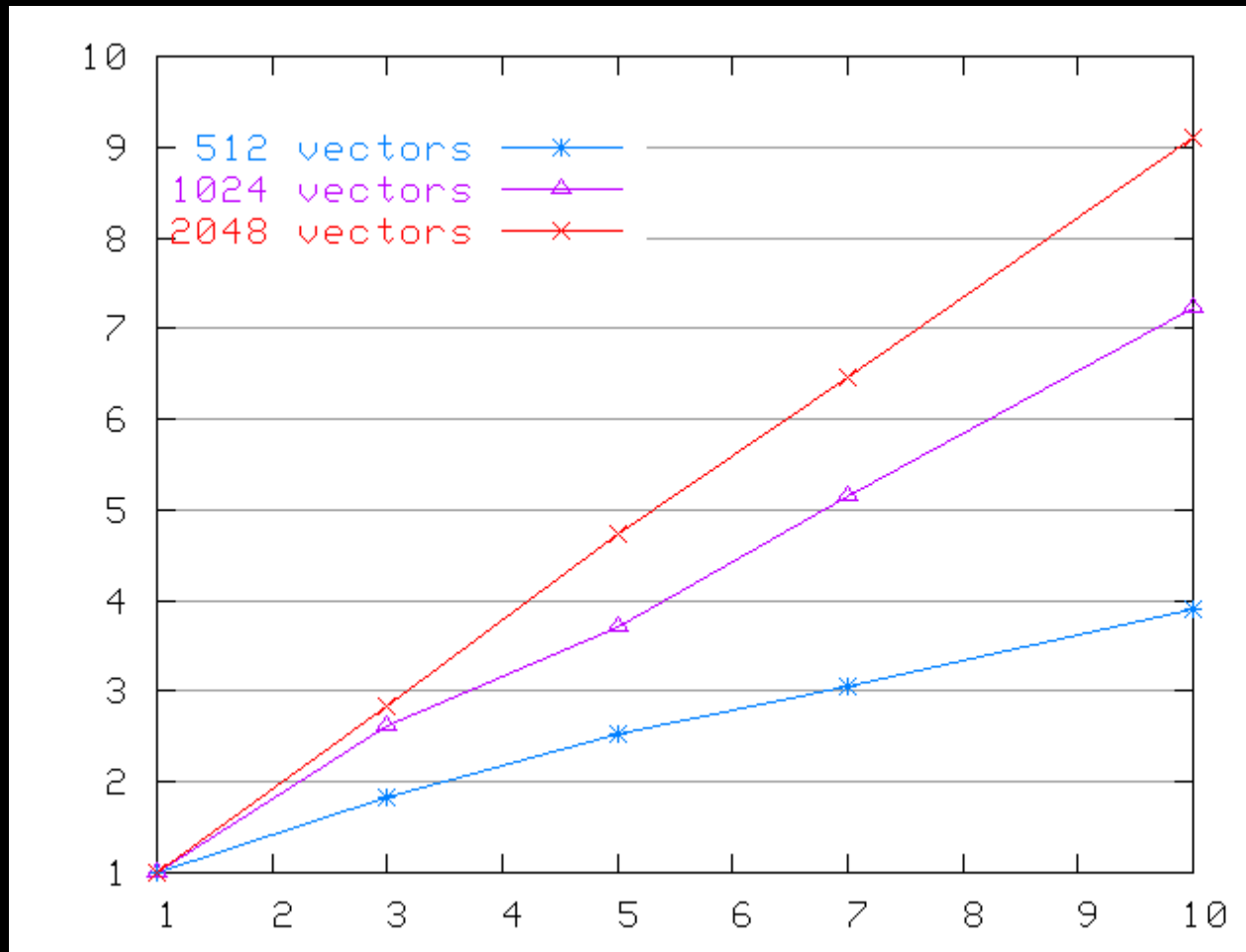


# Prototype : Vigne



# Speed-up of MGS

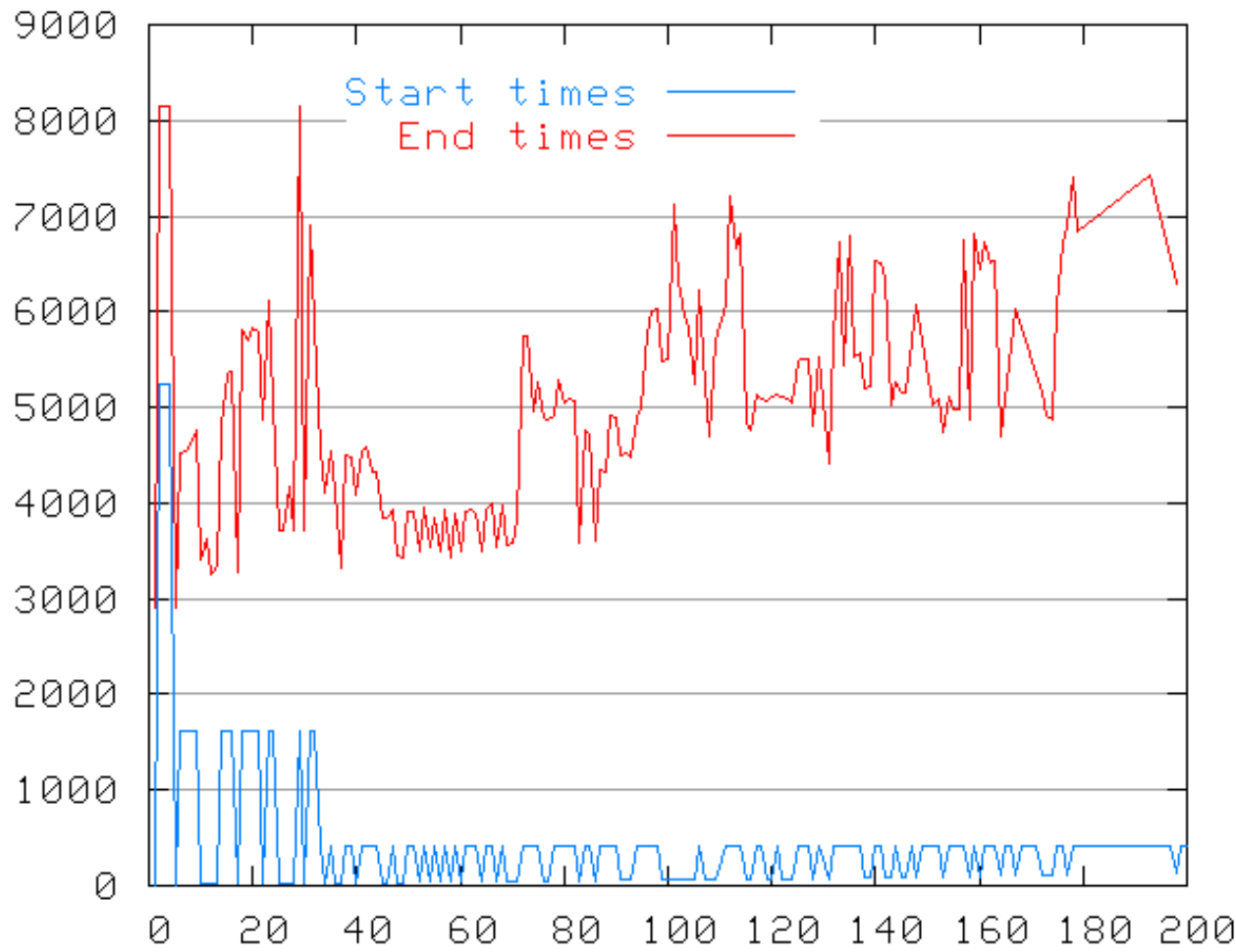
Speed-up



Number of nodes

# Simulation of a Highly dynamic Configuration

Time (s)



Number of consumers

Component 0:  
repeat 1000 times  
... write(x,v)  
barrier()  
barrier()

Component 1..n:  
repeat 1000 times  
barrier()  
read(x) ...  
barrier()

Impact of  
degraded routing

# Conclusion

# Conclusion

- Write-invalidate consistency protocol that tolerates multiple and simultaneous failures



Formally validated



Sensitive to the quality of key-based routing



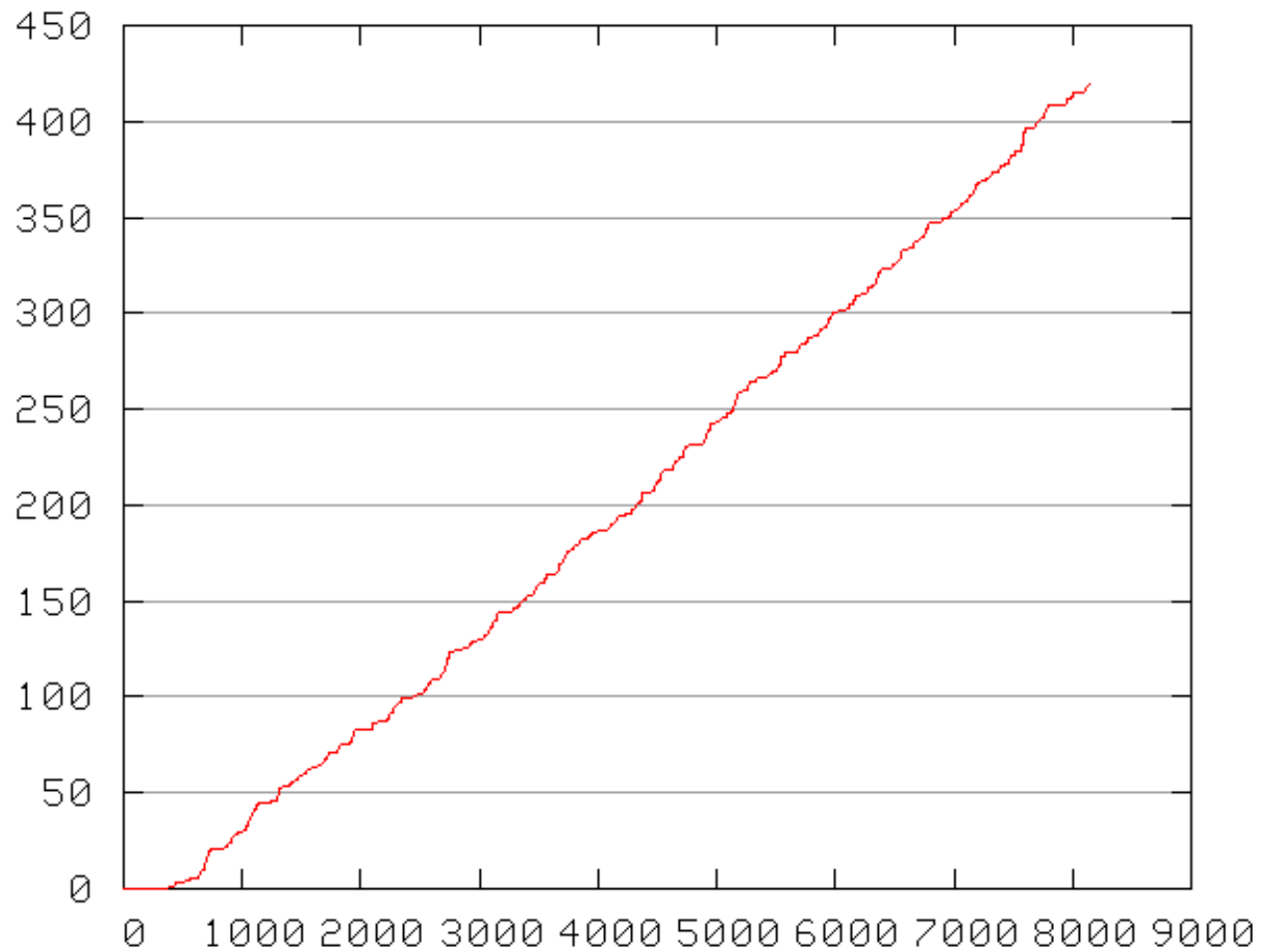
Encouraging performances

# Perspectives

- Lower the latency of requests
  - Fewer messages routed
  - Relax the synchronization between replica of object managers
- Experiment with “real world” applications
- Complete the single system image approach
  - Integrate with a resource allocator application deployment

Thank you!

# Experimental Results



# Experimental Results

