A Practical Transparent Data Sharing Service for the Grid

Louis Rilling, Christine Morin
IRISA / PARIS research group
Rennes, France

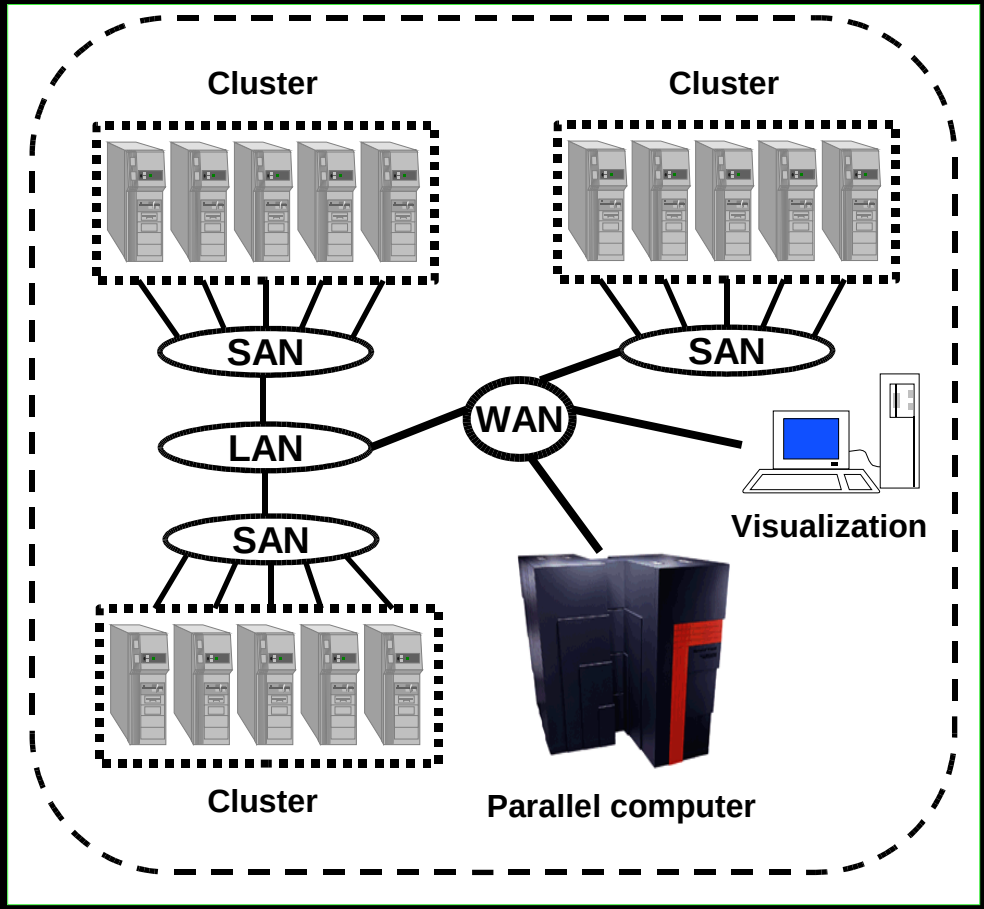
Objectives

- Distributed computing applications
- Why transparent data sharing should be fault-tolerant?

- How to obtain transparency and tolerance to multiple failures?

Challenges in Grids

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



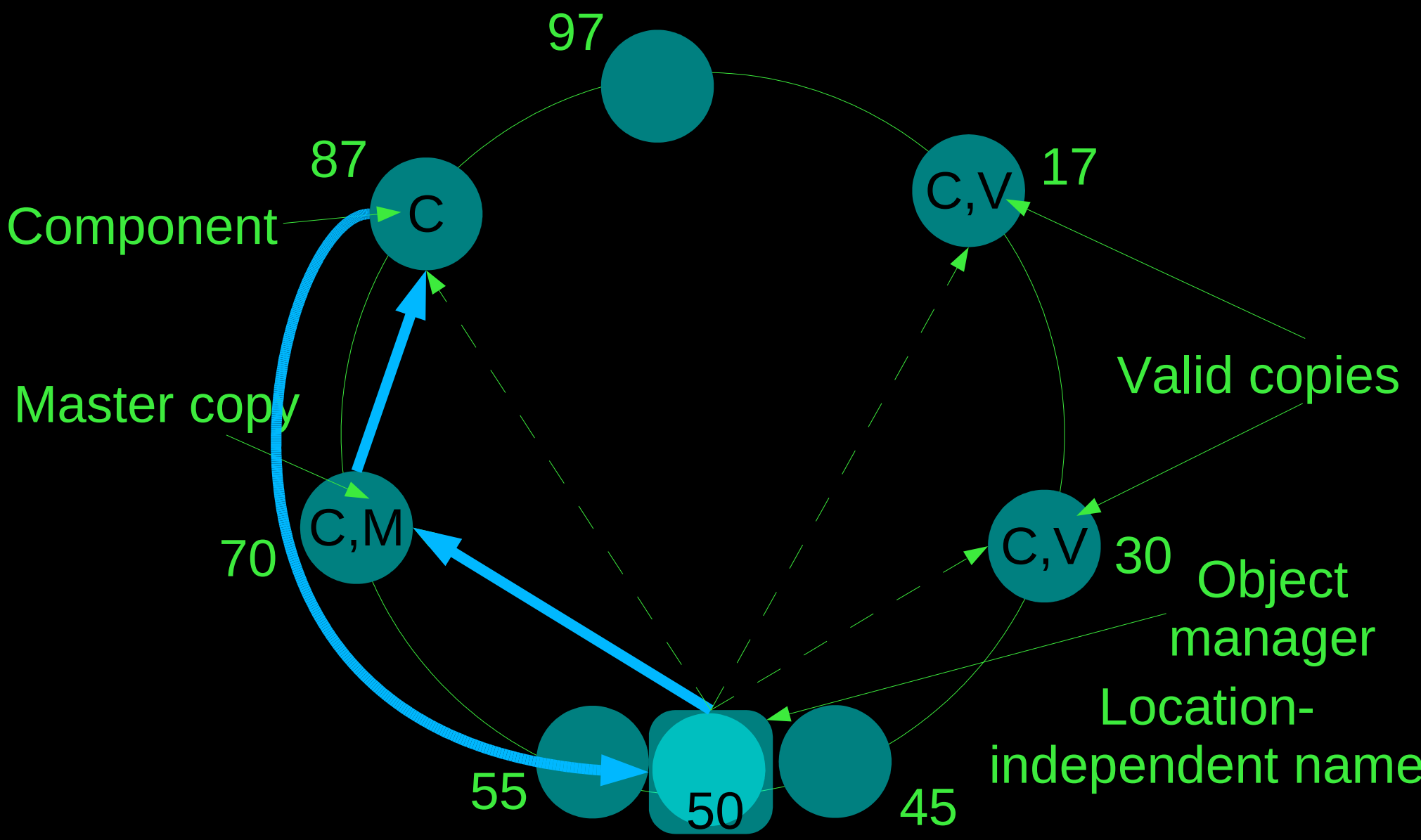
State of the Art

- 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

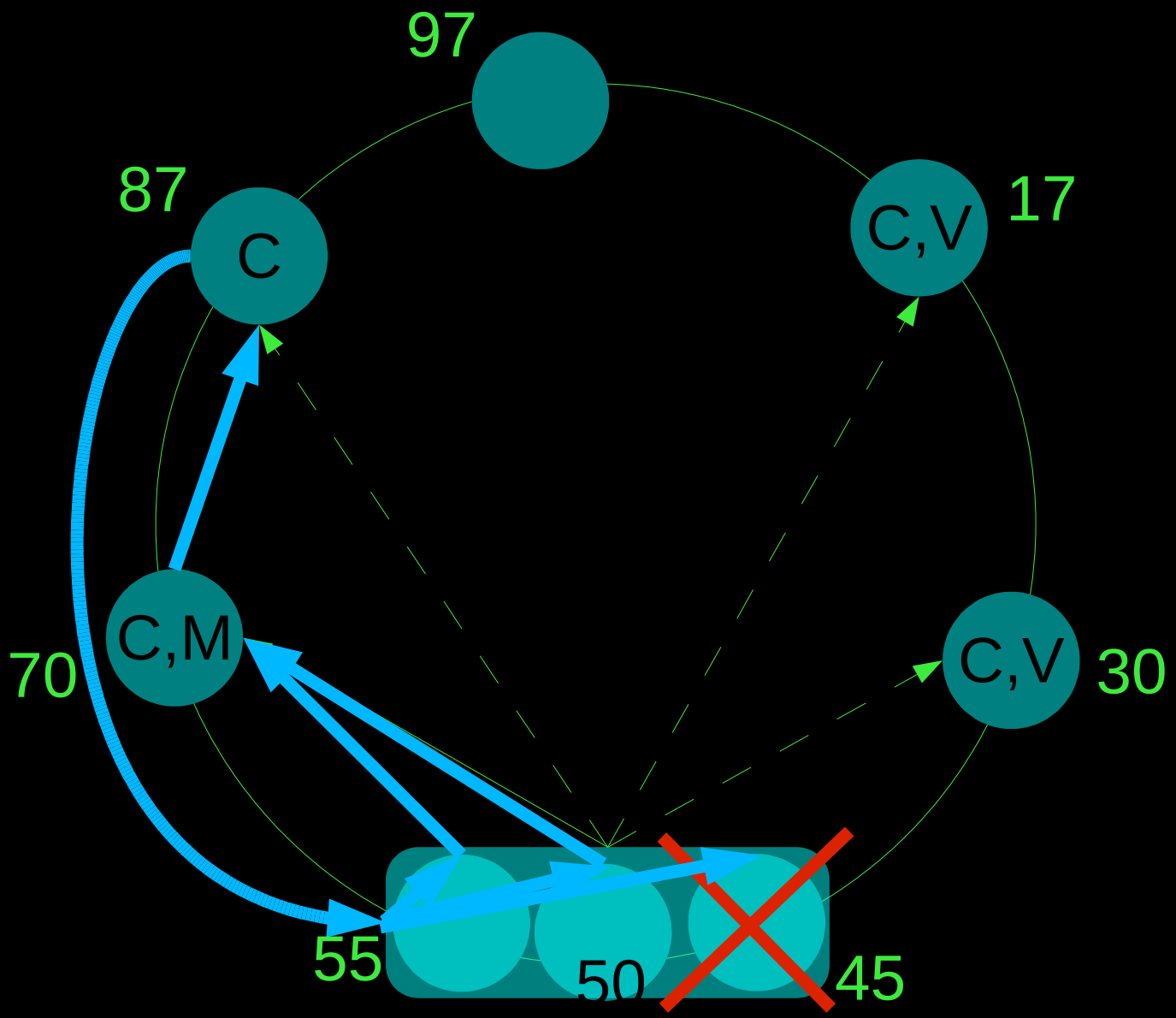
Method

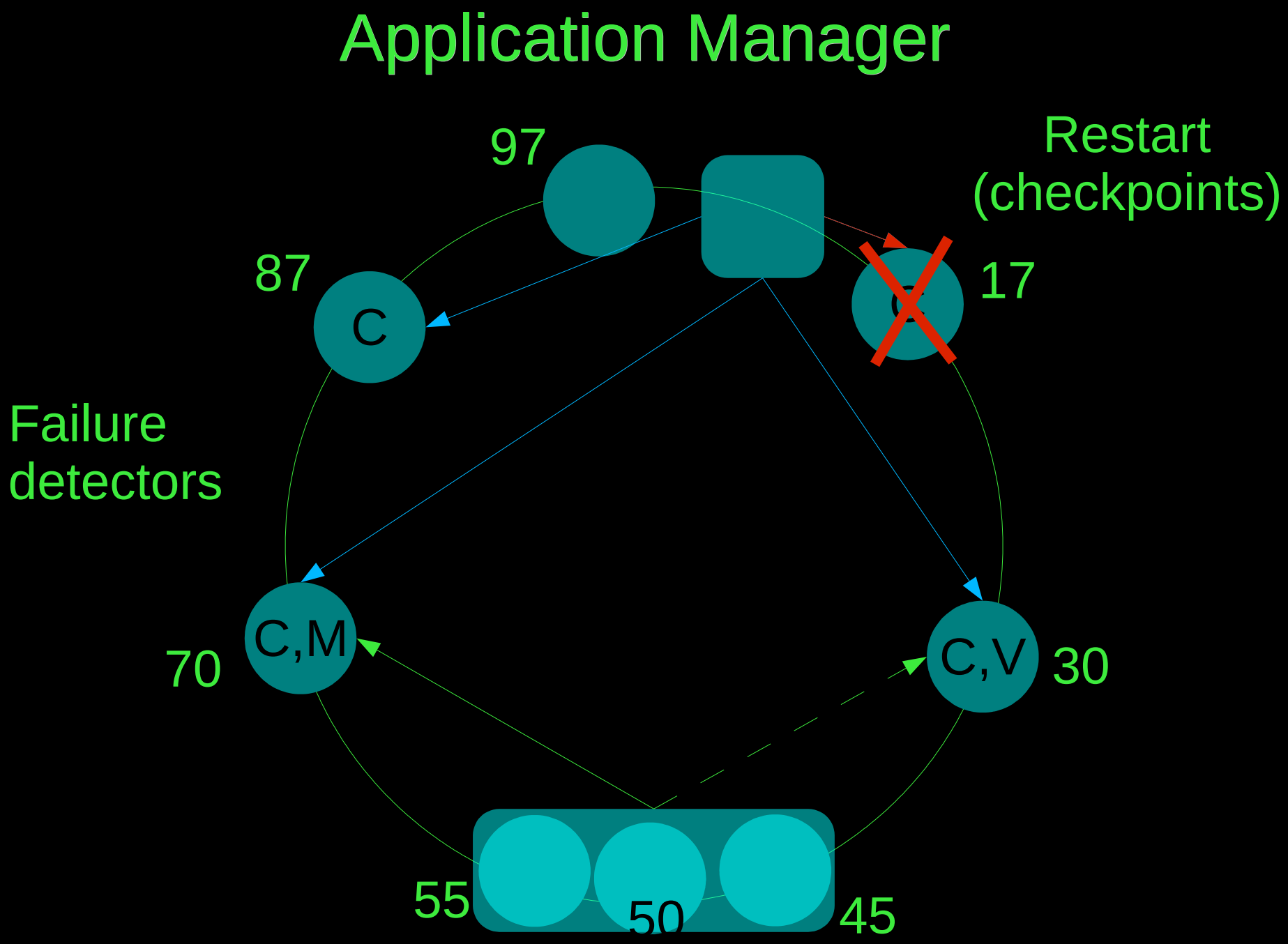
- Inspire from an existing consistency protocol
  - Write-invalidate
  - K. Li, static distributed managers
- Adapt the protocol to unreliable communications
  - Messages received in disorder, duplicated
- Tolerate multiple and simultaneous reconfigurations

Protocol for Atomic Consistency



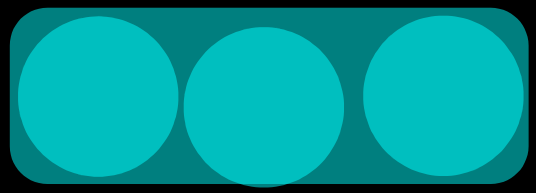
Object Manager Actively Replicated



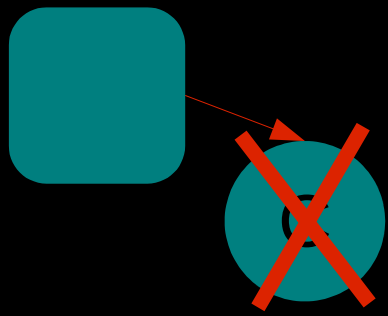

Application Manager
97
Restart
(checkpoints)
87
C
17
Failure
detectors
C,M
70
C,V
30
55
50
45

Summary

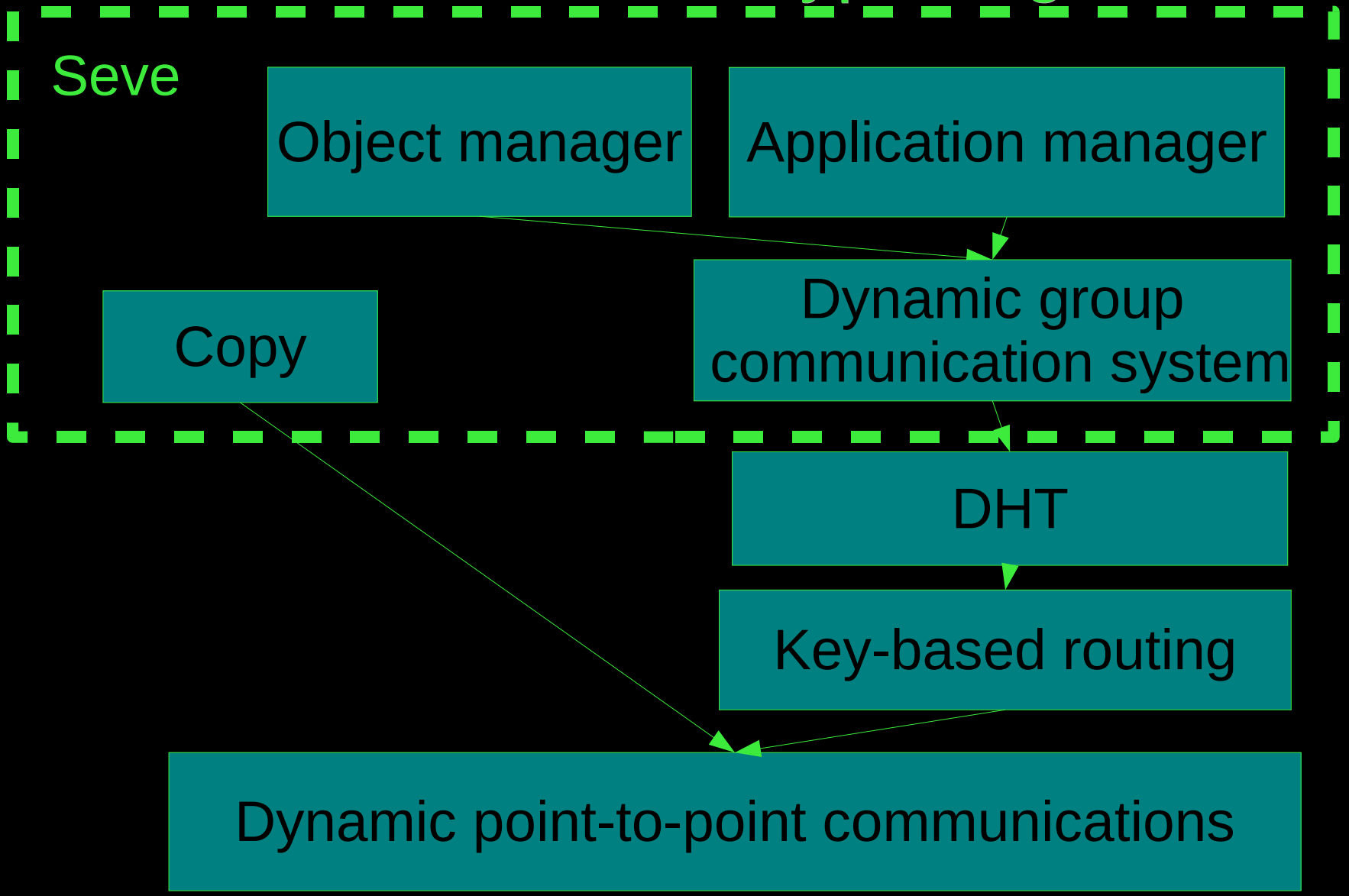
- Object manager in DHT
  - Active replication
  - Reconfiguration simplified



- Failure of an execution node -> restart (checkpoints) + application manager in DHT
  - Active replication

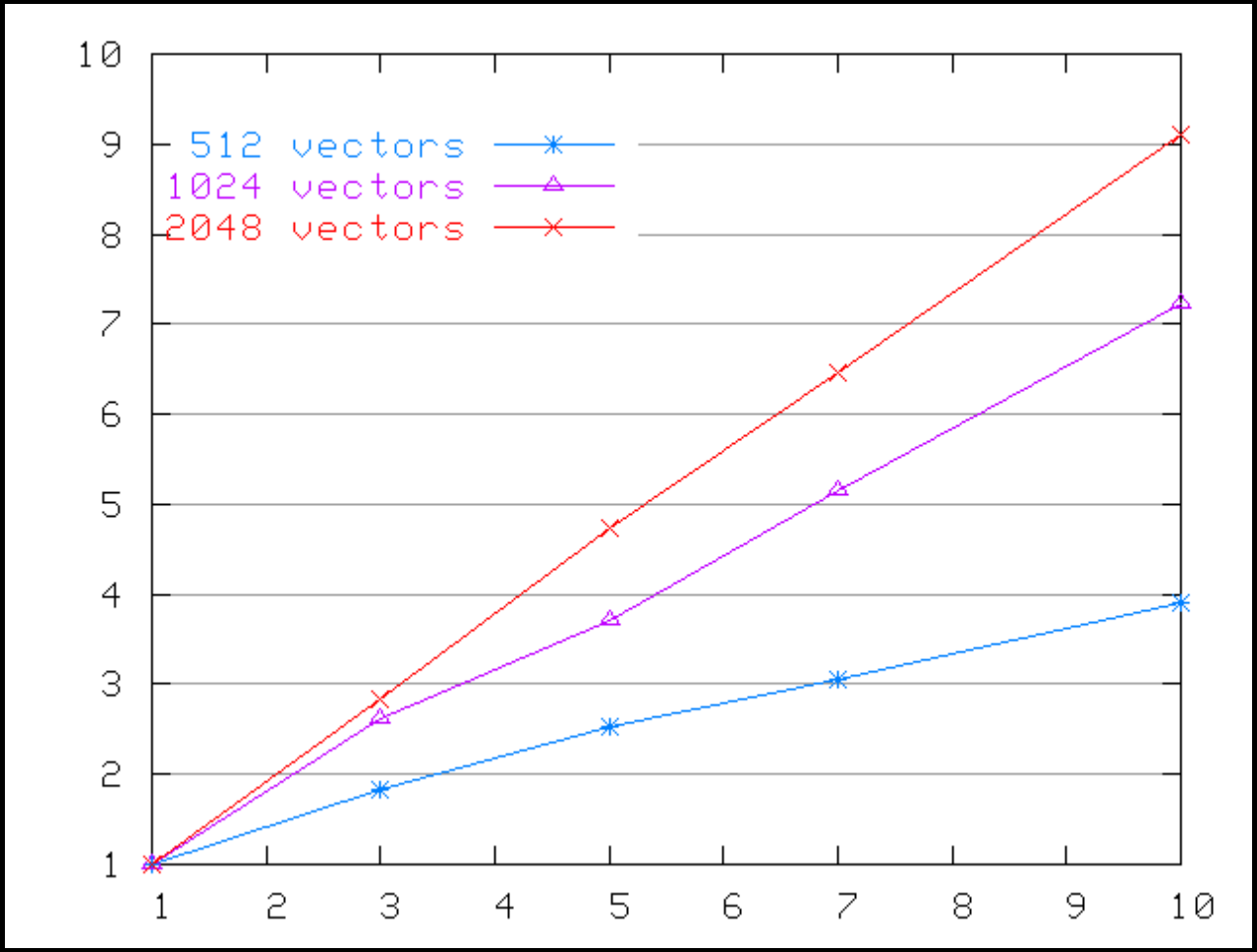


Runtime Prototype: Vigne



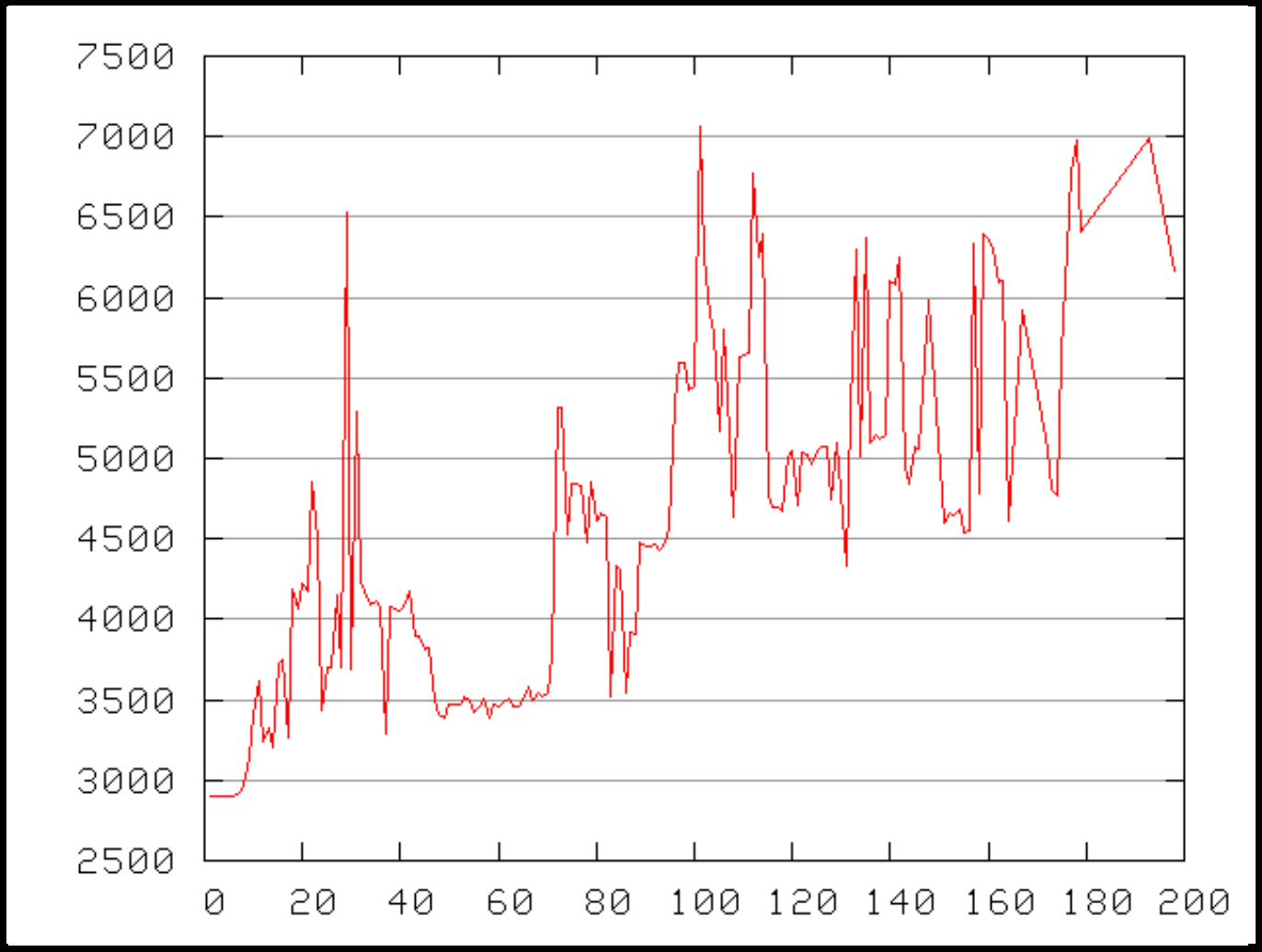
Speed-up of MGS

Speed-up



Number of nodes

Simulation of a Highly Dynamic Configuration



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
- Investigate other consistency models
- Experiment with “real world” applications
- Complete the single system image approach
  - Integrate with a ressource allocator
    application deployment

Thank you!