Recap.

- *Ubiquitous Computing Vision*
  - *The Computer for the Twenty-First Century*, Mark Weiser
  - *The Coming Age Of Calm Technology*, Mark Weiser and John Seely Brown
  - *People, Places, Things: Web Presence for the Real World* Tim Kindberg, John Barton, Jeff Morgan, Gene Becker, Ilja Bedner, Debbie Caswell, Phillipe Debaty, Gita Gopal, Marcos Frid, Venky Krishnan, Howard Morris, Celine Pering, John Schettino, Bill Serra.
  - *Next Century Challenges: Data-Centric Networking for Invisible Computing.* Mike Esler, Jeffrey Hightower, Tom Anderson, and Gaetano Borriello

Recap

- Distributed Systems Architecture
  - Intro. to Distributed system architecture (Domain Name Service (DNS), Gnutella, DNS round robin etc.)
  - *Oceanstore: An Extremely Wide-Area Storage System* David Bindel, Yan Chen, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Haim Weatherspoon, Westley Weimer, Christopher Wells, Ben Zhao, and John Kubiatowicz
  - *Feasibility of a Serverless Distributed File System Deployed on an Existing Set of Desktop PCs* William J. Bolosky, John R. Douceur, David Ely, and Marvin Theimer

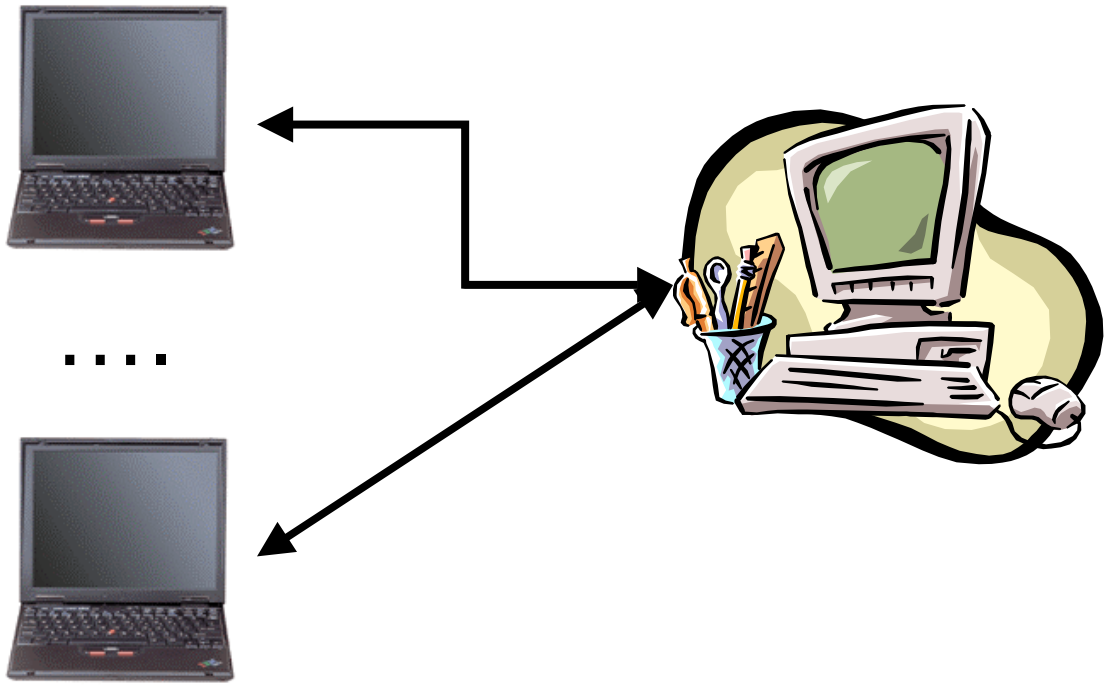
Recap

- Location and Naming management
  - *The Anatomy of a Context-Aware Application* Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, Paul Webster
  - *Active Names: Flexible Location and Transport of Wide-Area Resources* Amin Vahdat, Michael Dahlin, Thomas Anderson, and Amit Aggarwal

Outline

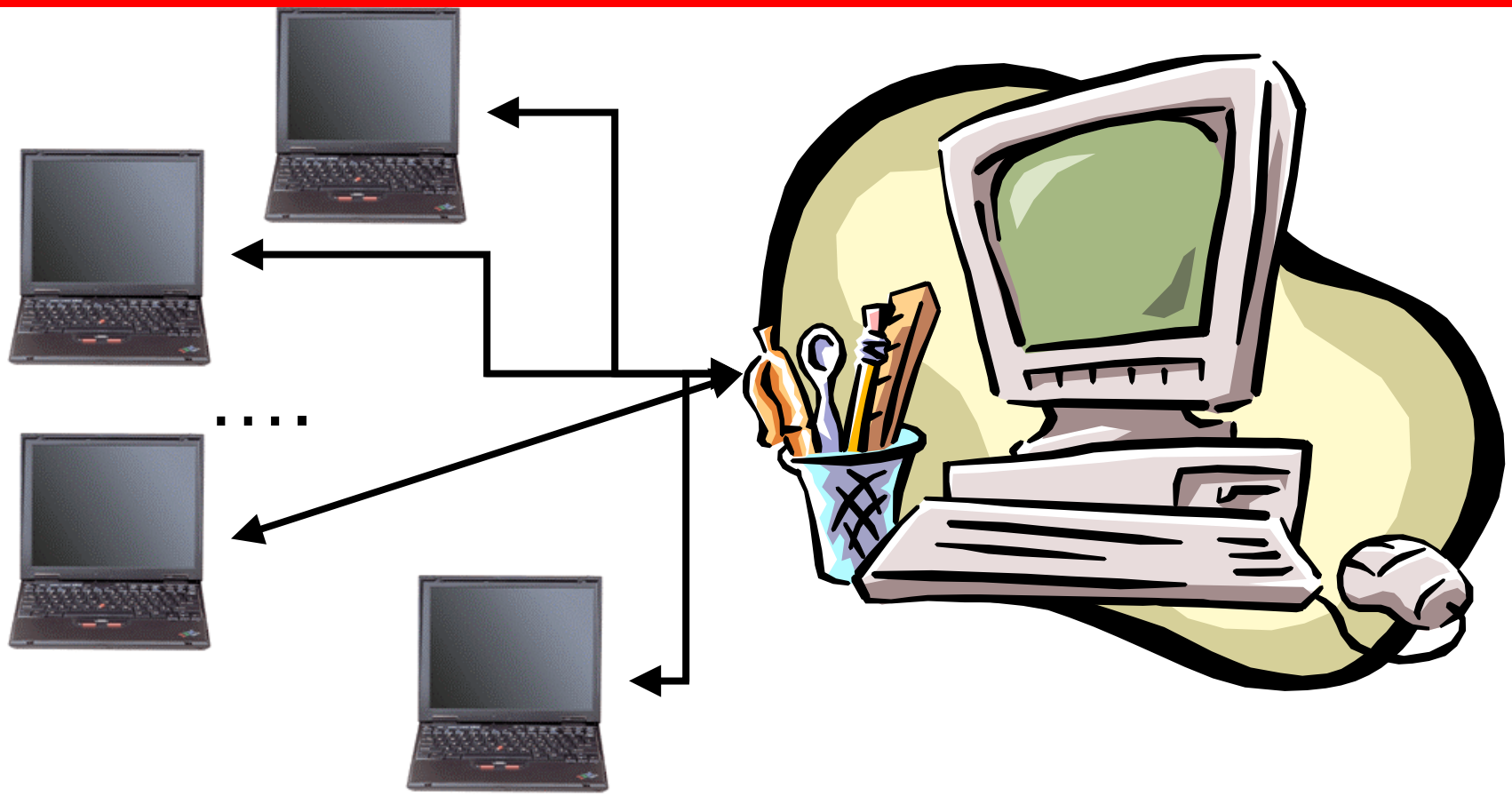
1. *The Dangers of Replication and a Solution, Jim Gray, Pat Helland, Patrick O'Neil, and Dennis Shasha. In Proceedings of the ACM SIGMOD international conference on Management of data, 1996*

Replication – Intro.



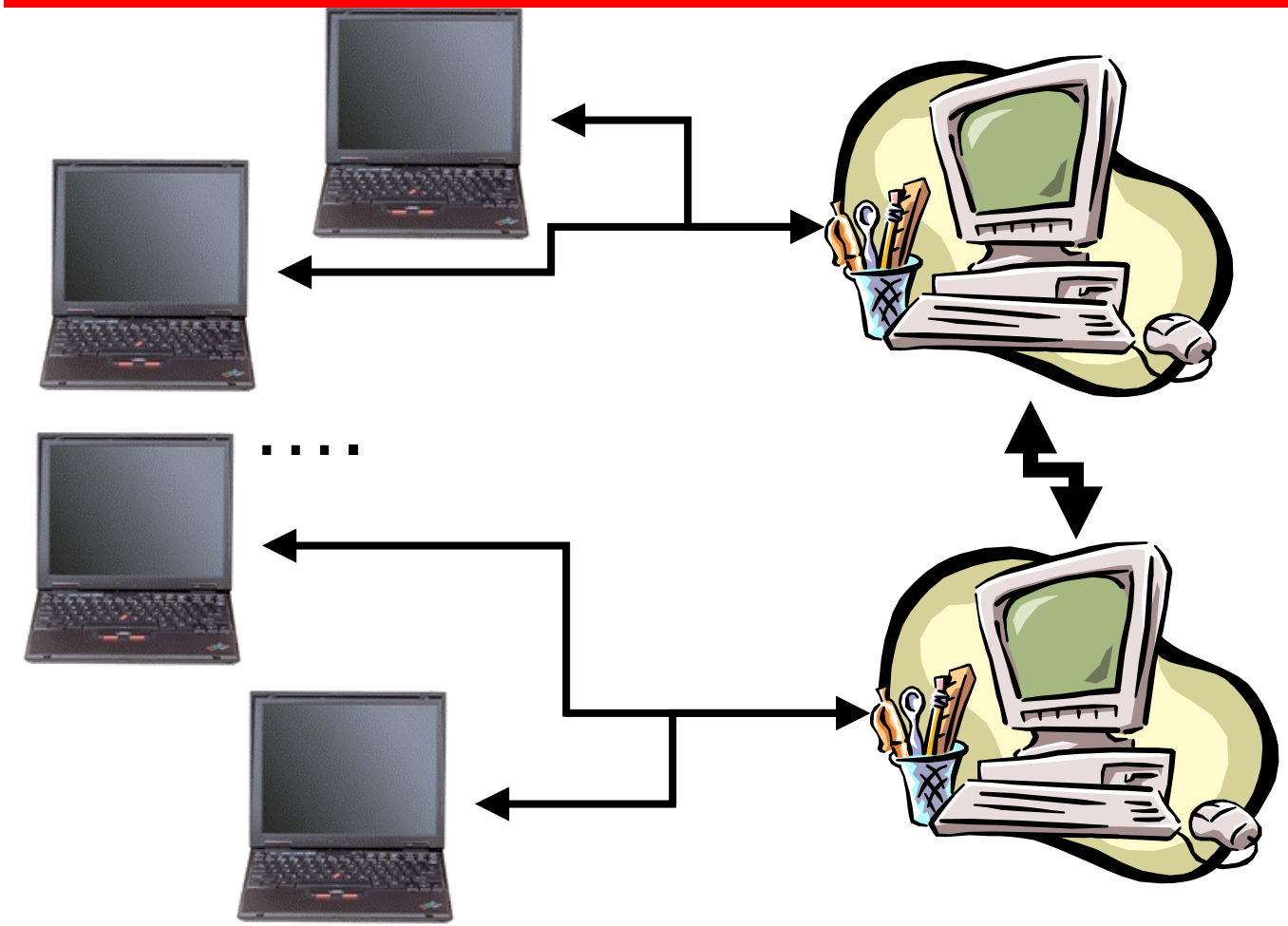
- As systems grow, need to scale up

Scale Up



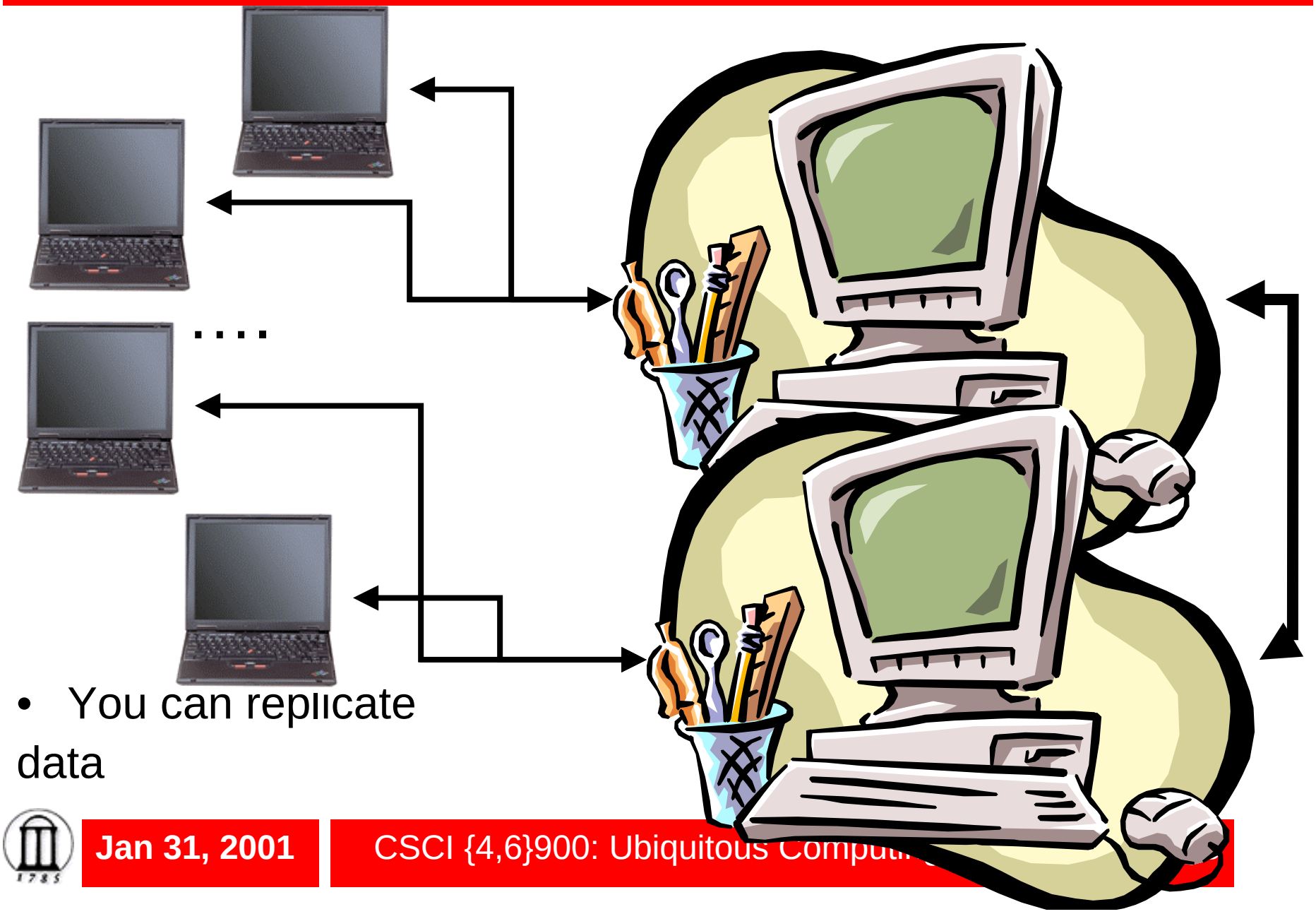
- You can scale up by buying a bigger machine

Partition



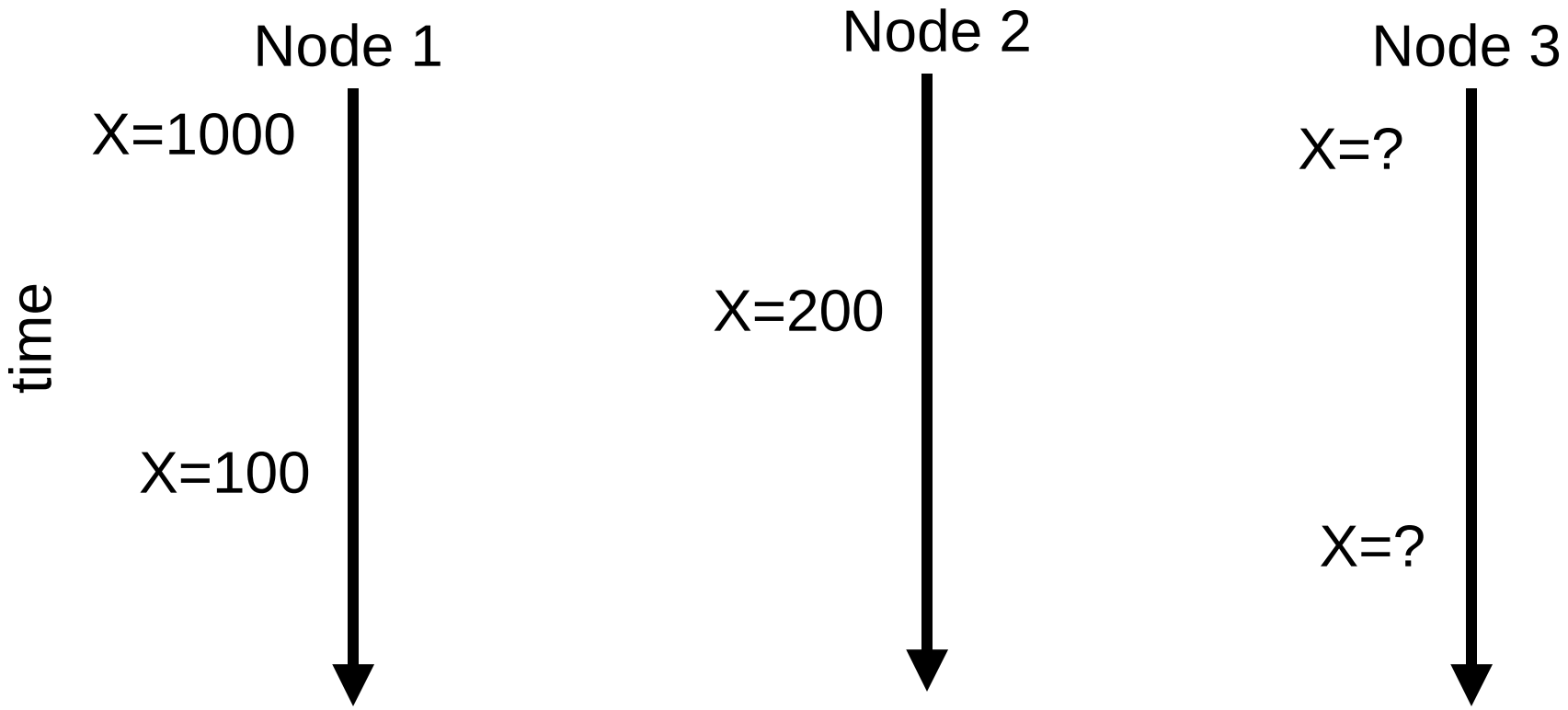
- You can scale up by partitioning the machines (e.g. service users in east coast from Atlanta and west coast from L.A.)

Replication



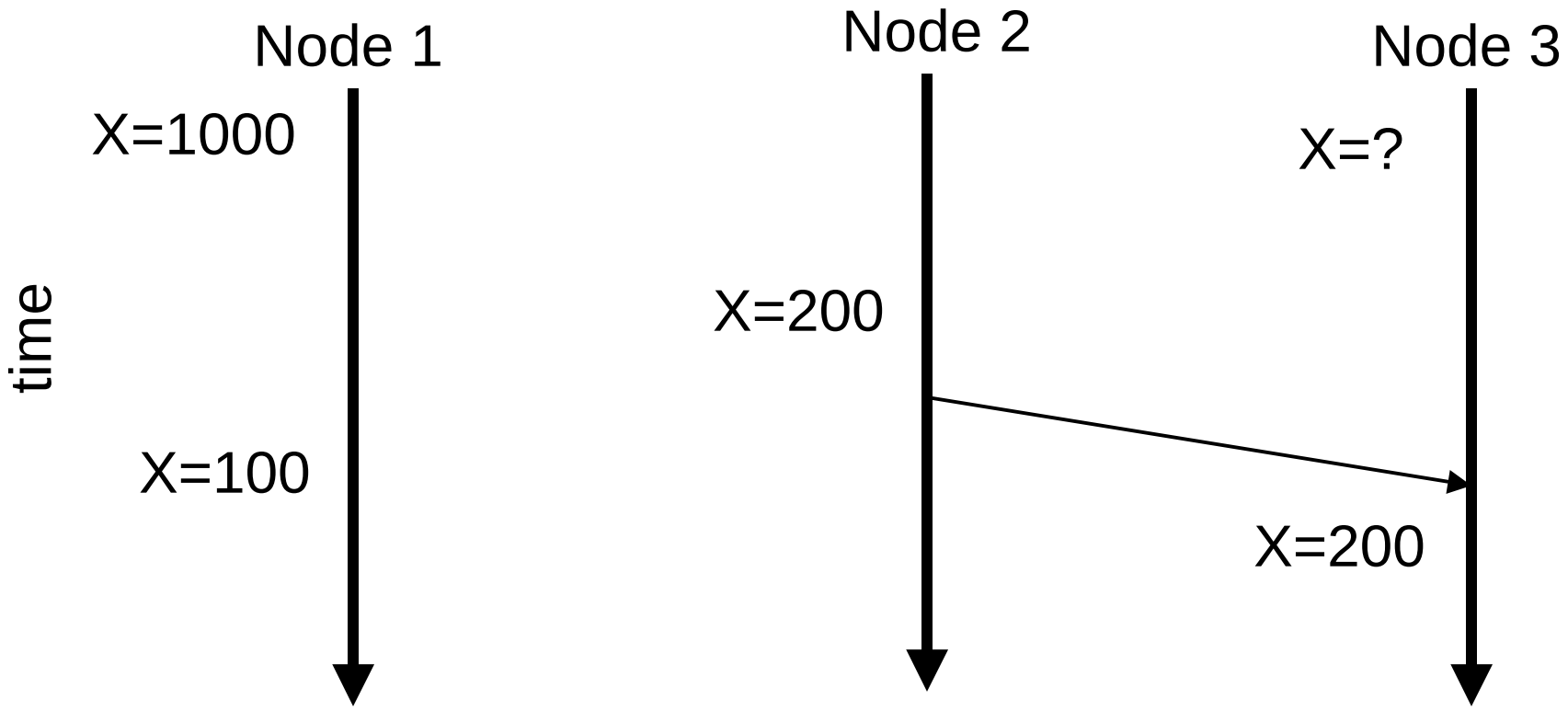
- You can replicate data

Serializability – Intro.



- What is the value of X in node 3?
- Causal ordering (Update x when you hear from Node 1 or Node 2)

Serializability – Intro.



- What is the value of X in node 3?
- Causal ordering (Update x when you hear from Node 1 or Node 2)

Goals of replication

- Availability and scaleability

  Provide high availability and scaleability through replication

- Mobility

  Allow mobile nodes to read and update the database while disconnected from the network

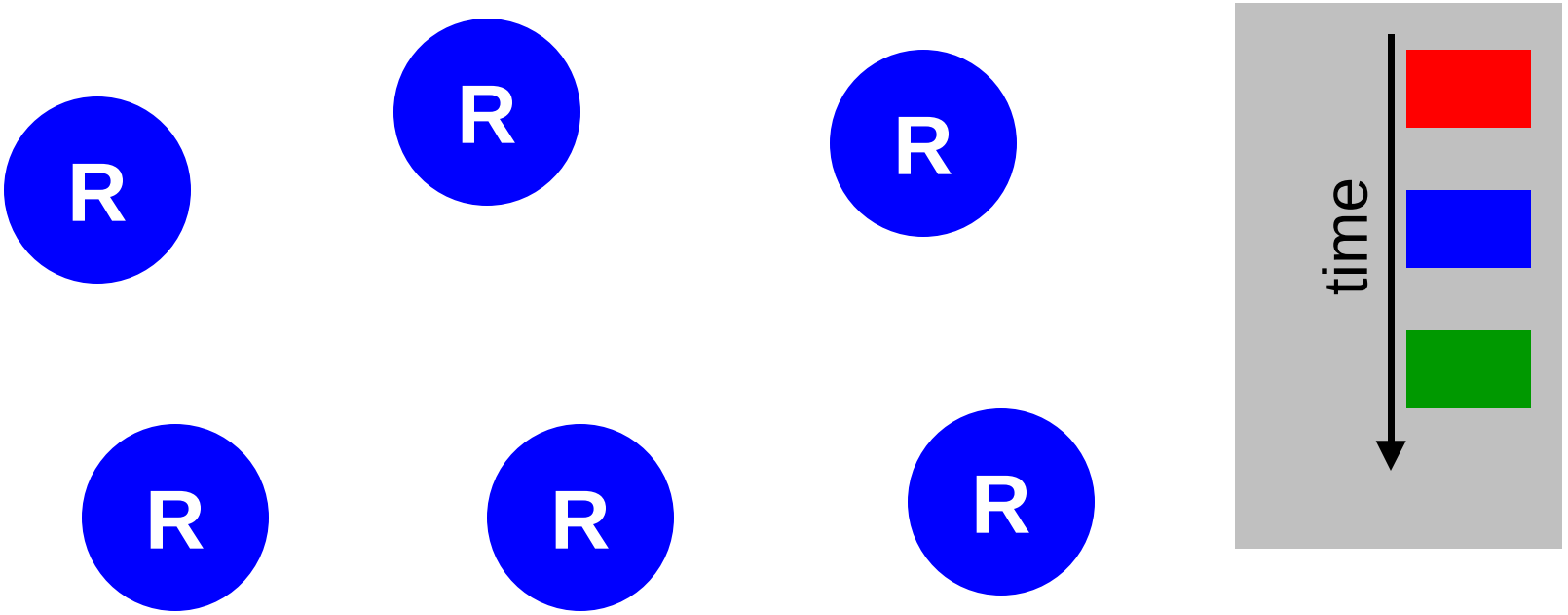
- Serializability

  Provide single-copy serializable transaction execution

- Convergence

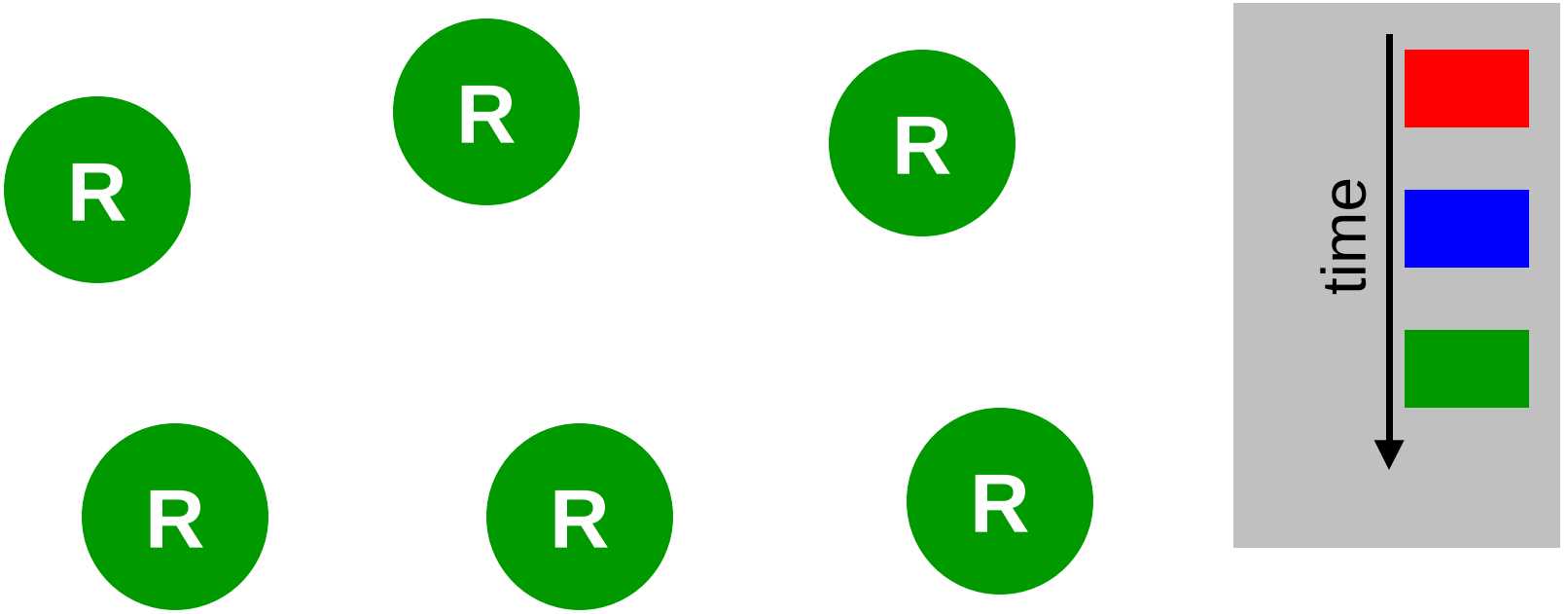
  Provide convergence to avoid system delusion

Eager Replication



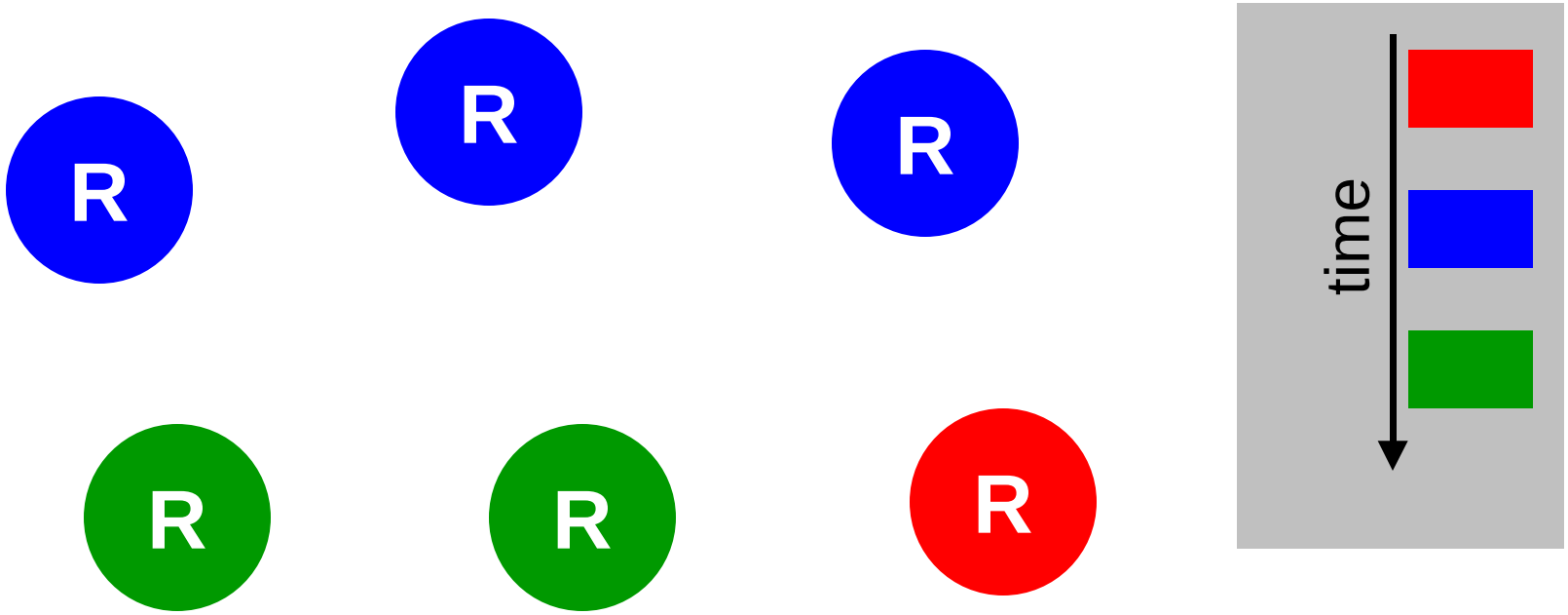
- All replicas synchronized to the same value immediately

Eager Replication



- All replicas synchronized to the same value
- Lower update performance and response time

Lazy Replication



- One replica is updated by the transaction
- Replicas synchronize asynchronously
- Multiple versions of data

Single node Transaction

Checking –1000
Savings +500
CD +500
Commit

- No conflicts

Eager Transaction

Checking –1000

Savings +500

CD +500

Commit

Checking –1000

Savings +500

CD +500

Commit

Checking –1000

Savings +500

CD +500

Commit

N nodes – N times as much work

Lazy Transaction

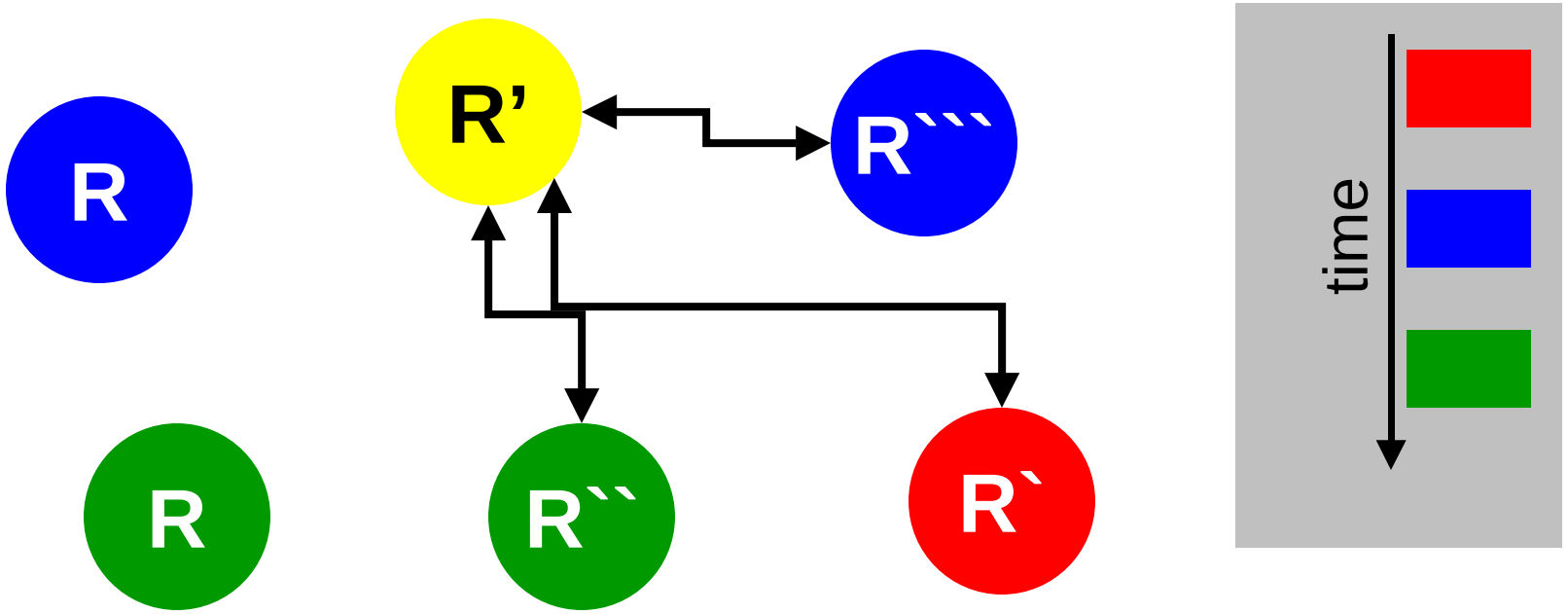
Checking –1000
Savings +500
CD +500
Commit

Checking –1000
Savings +500
CD +500
Commit

Checking –1000
Savings +500
CD +500
Commit

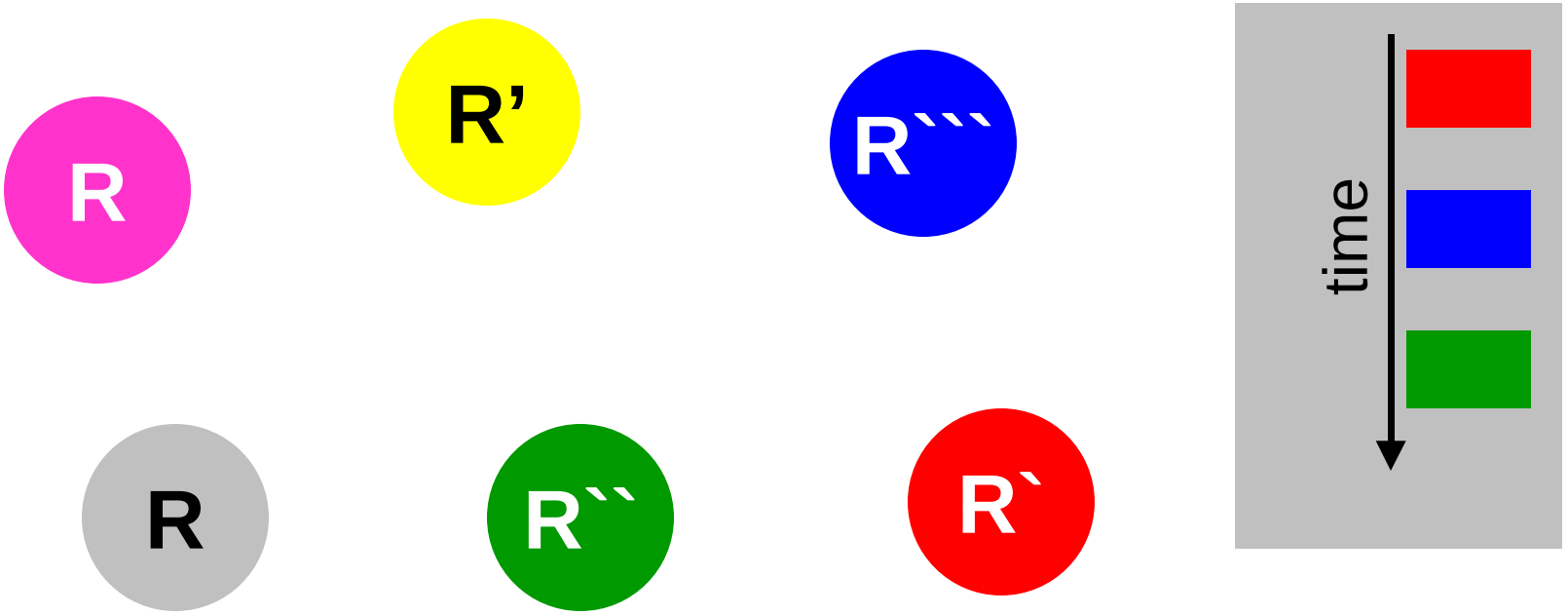
- N nodes – N times as much work
- N transactions

Concurrency anomaly in Lazy Replication



- R` - Which version of data should it see?
- If committed transaction is ‘wrong’, conflict
- Conflicts have to be reconciled

Scaleup pitfall



- When the nodes divulge hopelessly
- System delusion – database is inconsistent and no obvious way to repair it

Regulate replica updates

- Group: Any node with a copy can update item
  - Update anywhere

- Master: Only a master can update the primary copy. All replicas are read-only. All update requests are sent to the master

Replication strategies

Propagation Vs. Ownership	Lazy	Eager
Group	N transactions N object owners	1 transaction N object owners
Master	N transactions 1 object owner	1 transaction 1 object owner
Two tier	N+1 transactions, 1 object owner Tentative locate update, eager base update