In Search of an Understandable Consensus Algorithm - Raft

Diego Ongaro, John Ousterhout (Authors)

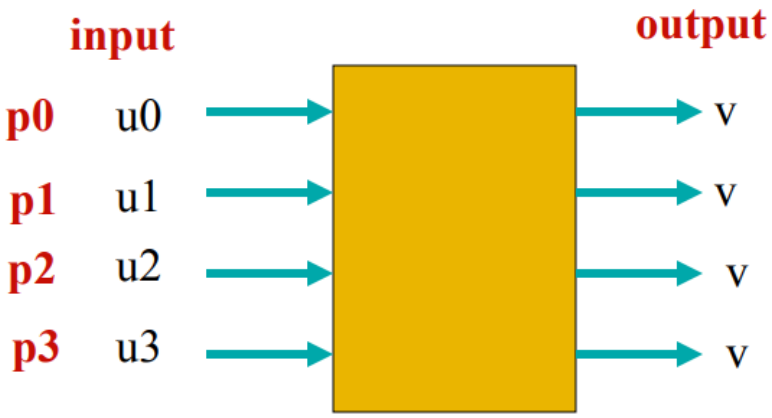
董峻铎 (Reporter)

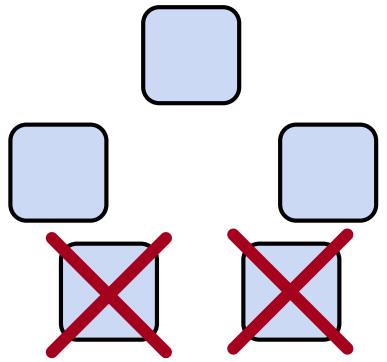
China University of Geosciences

https://raft.github.io/raft.pdf

What is Consensus?

- **involves multiple servers agreeing on values**
- **Recovers from server failures autonomously**
  - Minority of servers fail: no problem
  - Majority fail: lose availability, retain consistency



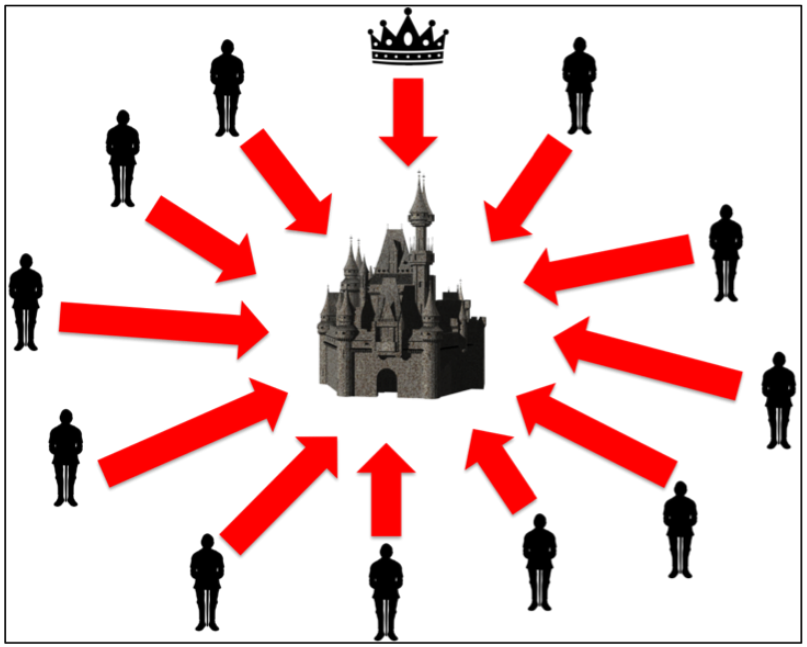


Servers

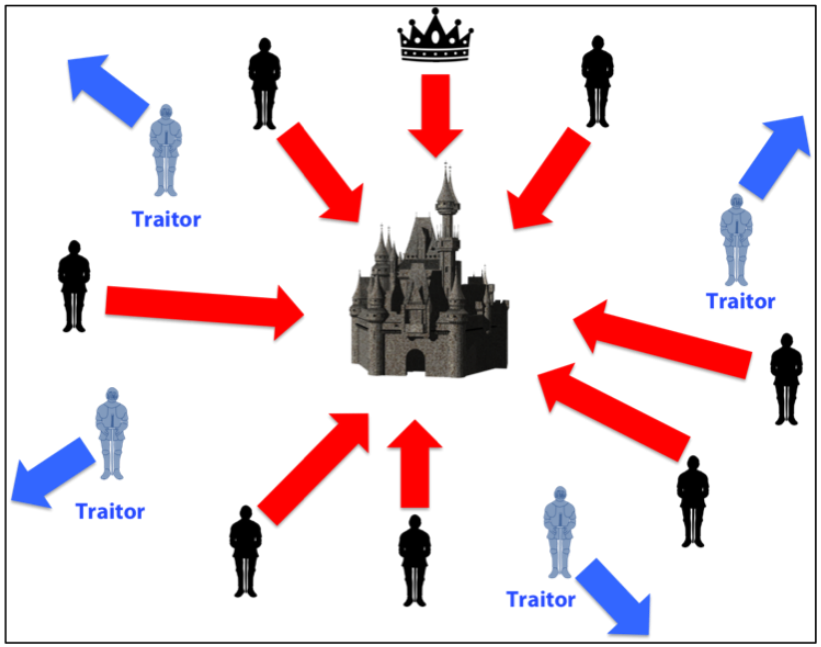
- **Key to building consistent storage systems**

Paxos? Too complicated!

- *Leslie Lamport, 1989*
- **Byzantine Generals Problem**



Coordinated Attack Leading to Victory



Uncoordinated Attack Leading to Defeat

- **Paxos: a consensus algorithm to solve it**
  - But… too difficult to be understood & implemented

Method: Design a new algorithm

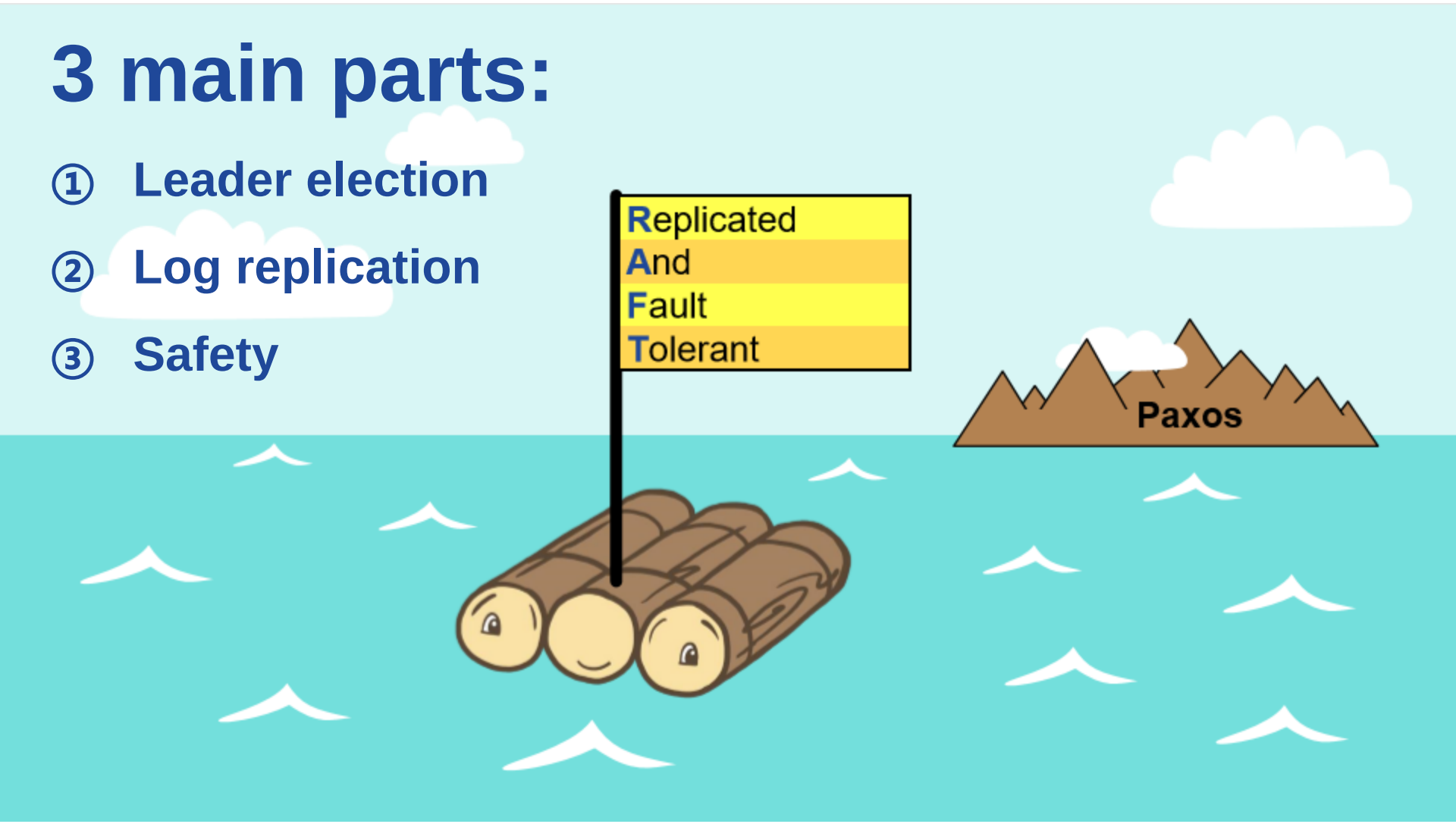
- **Named Raft**
- **Design for Understandability**
- **Define a better algorithm for building real systems**
  - Must be correct, complete, and perform well
  - Must also be understandable
- **Be easier to understand than Paxos**
  - Fundamentally different decomposition than Paxos
  - Less complexity

3 main parts:

① Leader election

② Log replication

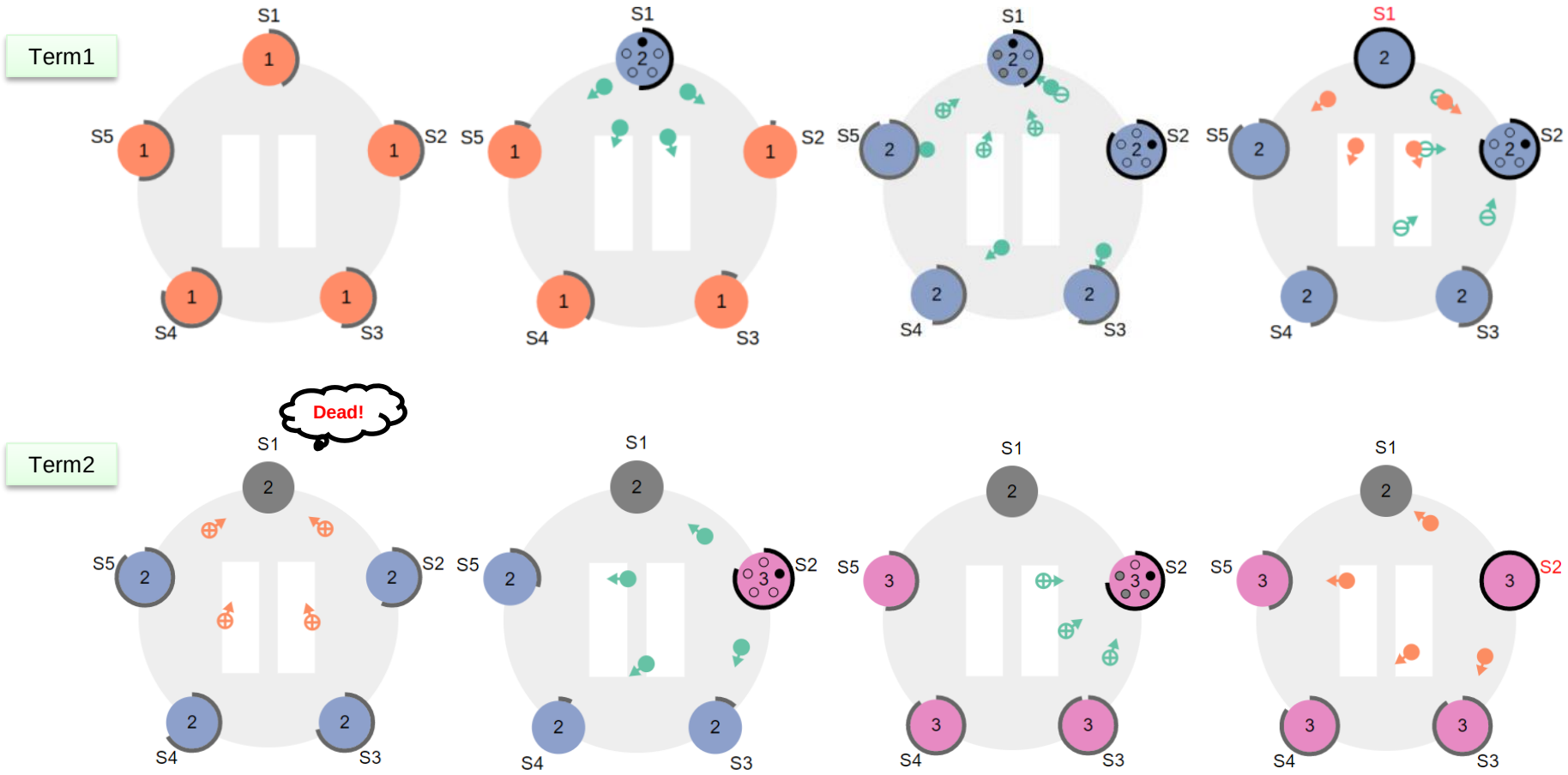
③ Safety



1. Leader election

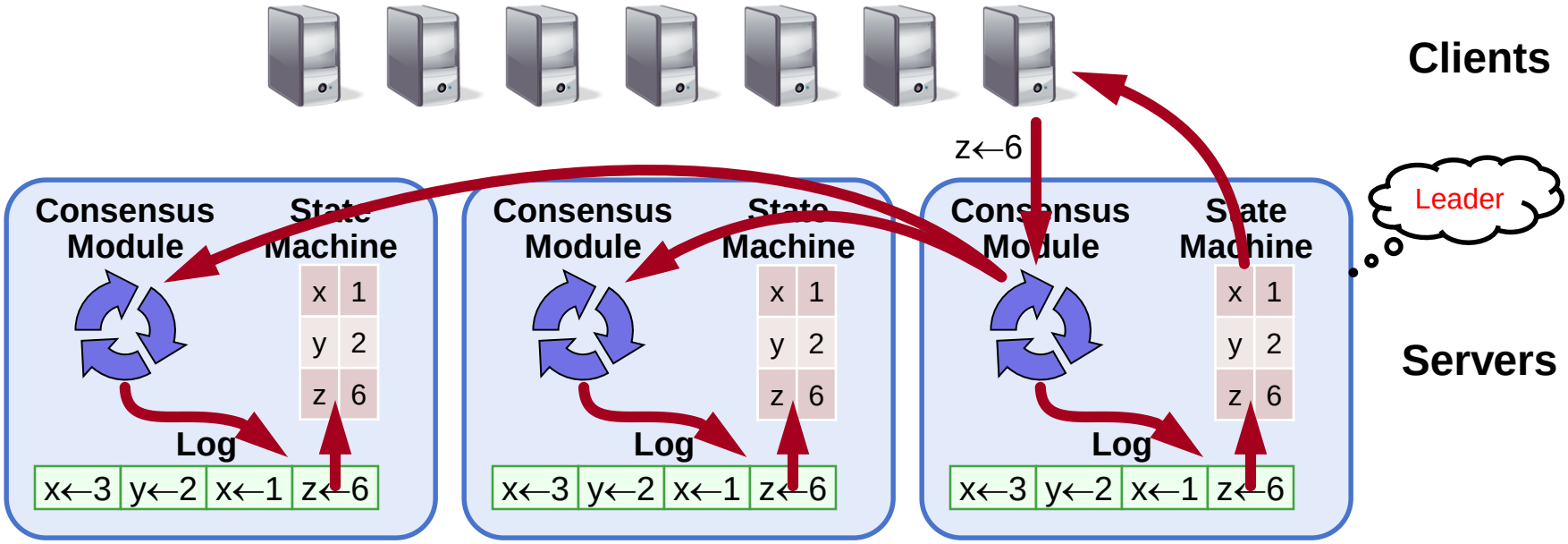
- **Select one of the servers to act as cluster leader**
- **If the leader is crashed, choose a new one**

$\geq \frac{1}{2}$ votes



2. Log replication

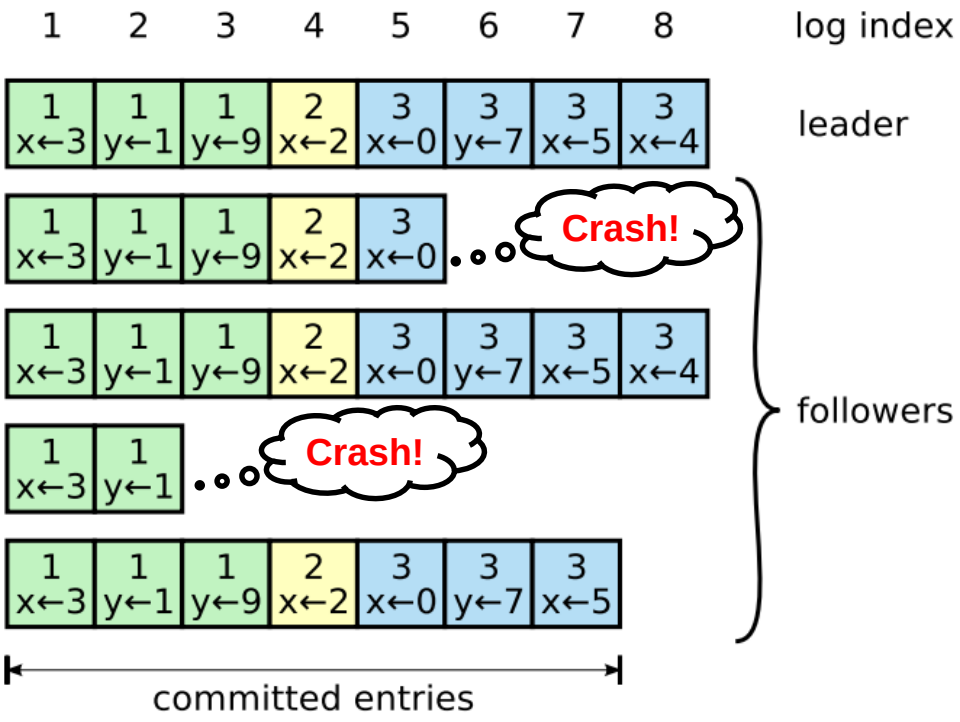
- **Replicated State Machines**



- **Replicated log ⇒ replicated state machine**
  - All servers execute same commands in same order
- **Consensus module ensures proper log replication**
- **Node crashes / Leader changes ⇒ replication may be wrong!**

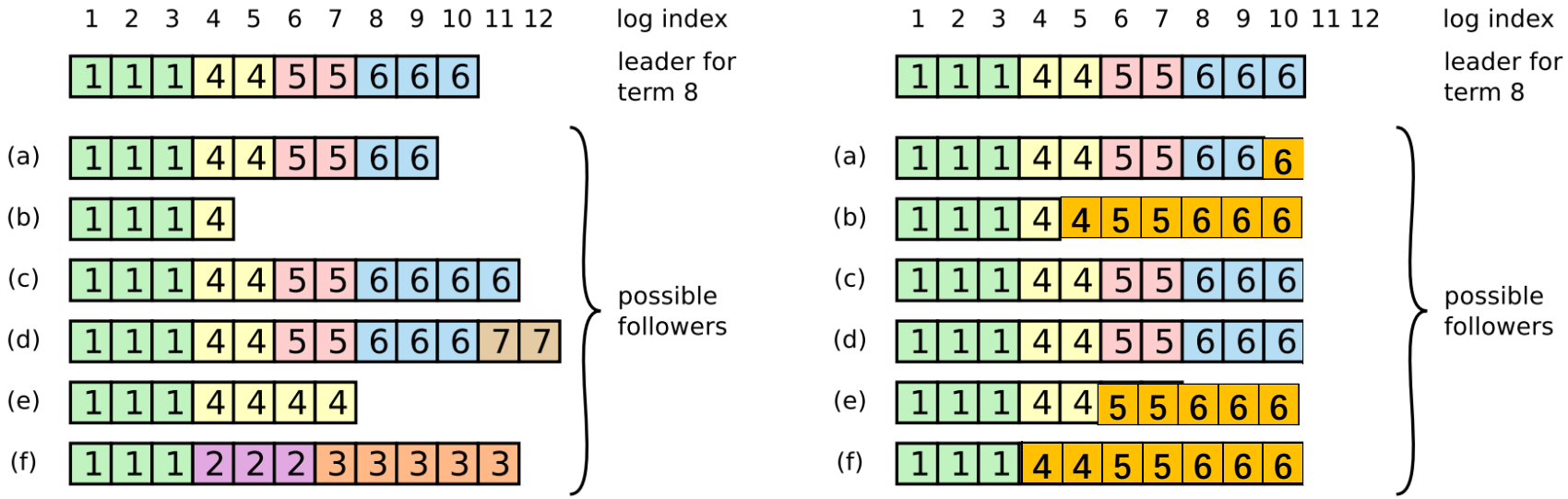
2. Log replication

- **Node crashes**
- **Safely replication**
  - More than half of nodes have replicated logs
  - then leader commits



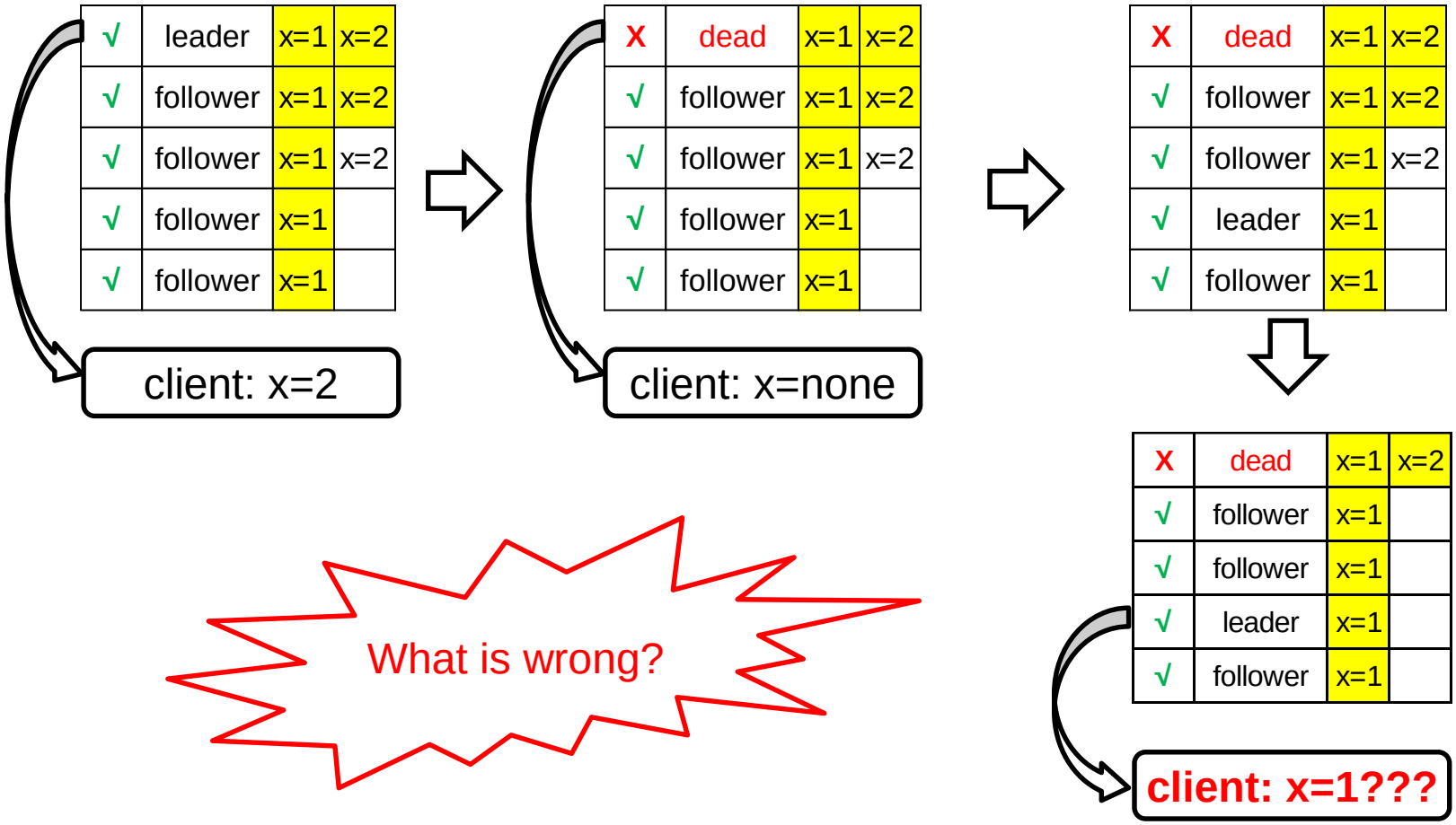
2. Log replication

- **Leader changes**
- **Find the last common log from the end to the front**
- **Force followers to replicate different logs!**



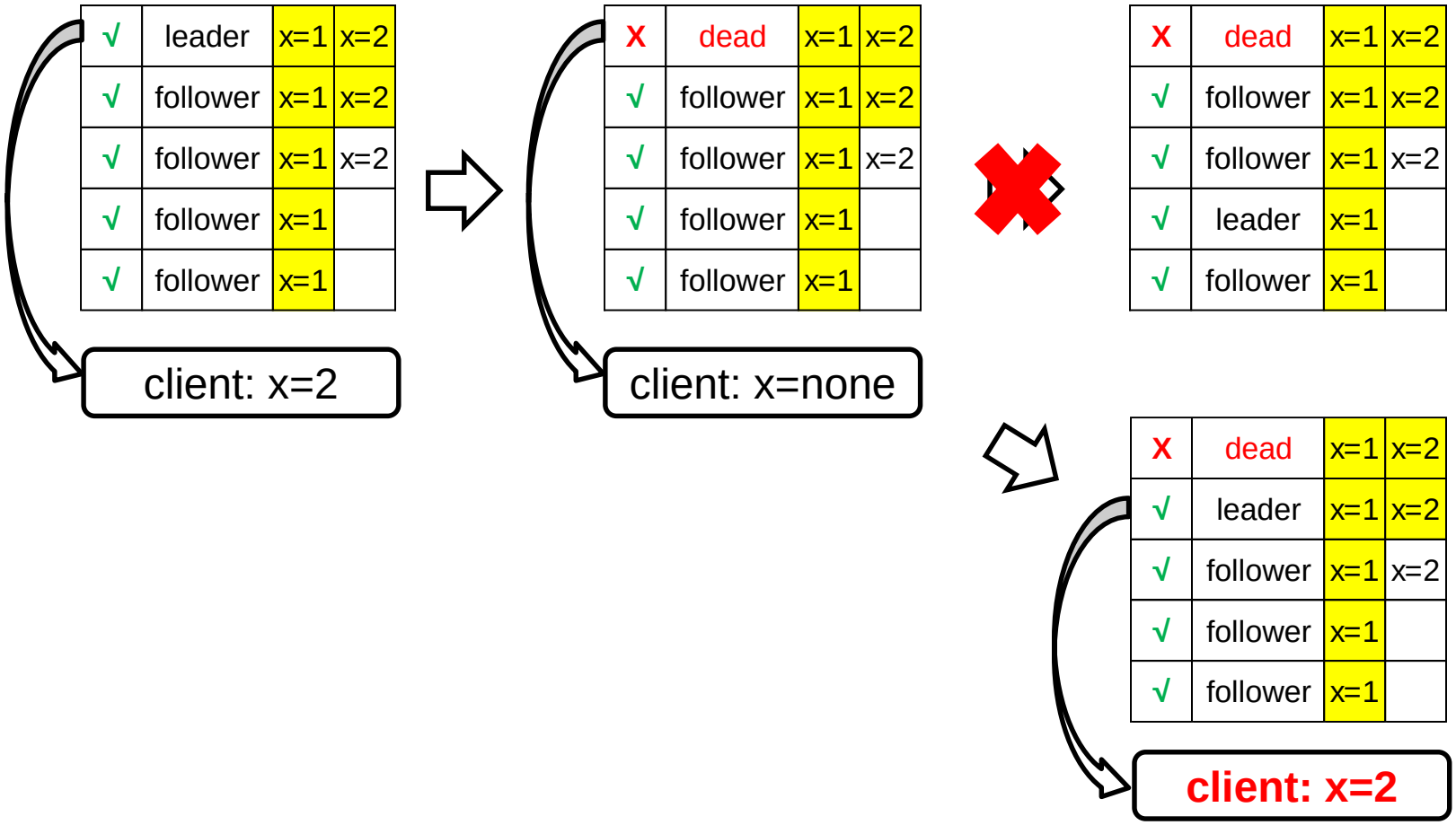
3. Safety

- **Hidden error of execute**
  - each node executes the same commands in the same order?



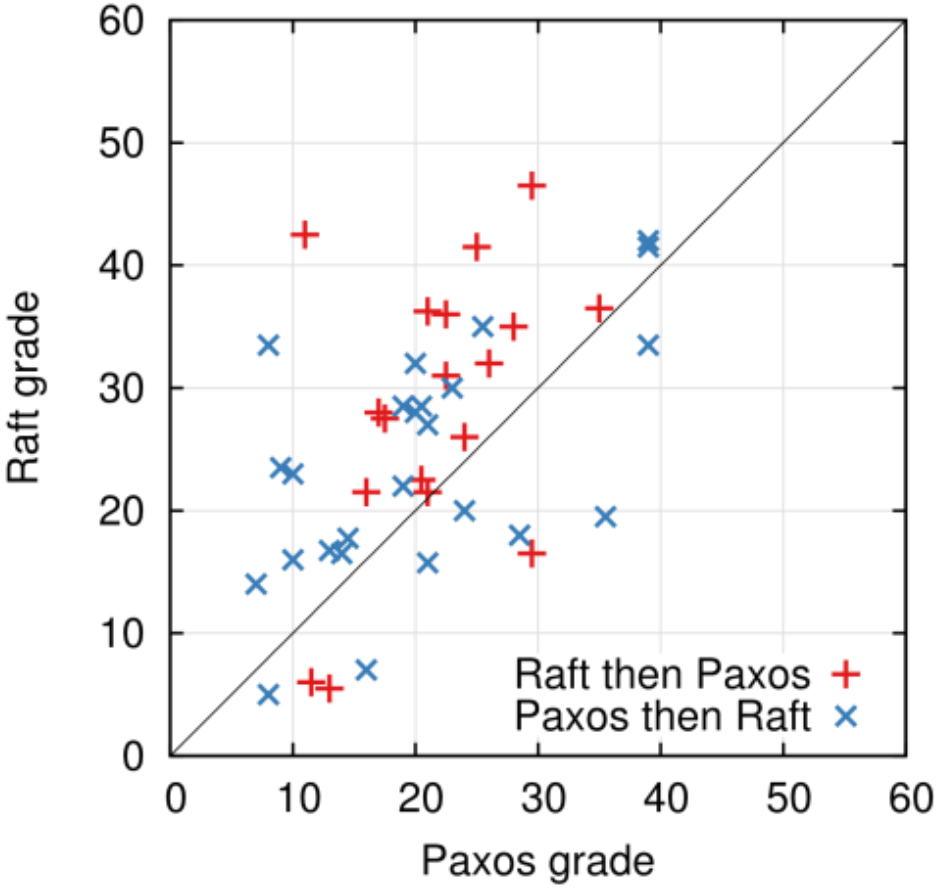
3. Safety

- **Node with <u>smaller committed log number</u> cannot be elected as leader**

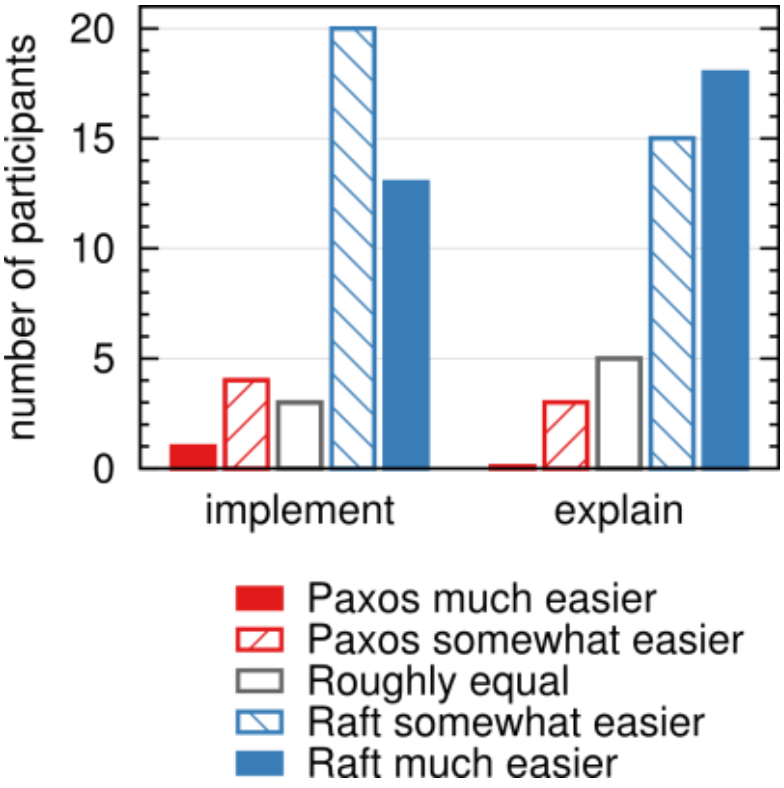


Result: Raft User Study

Quiz Grades

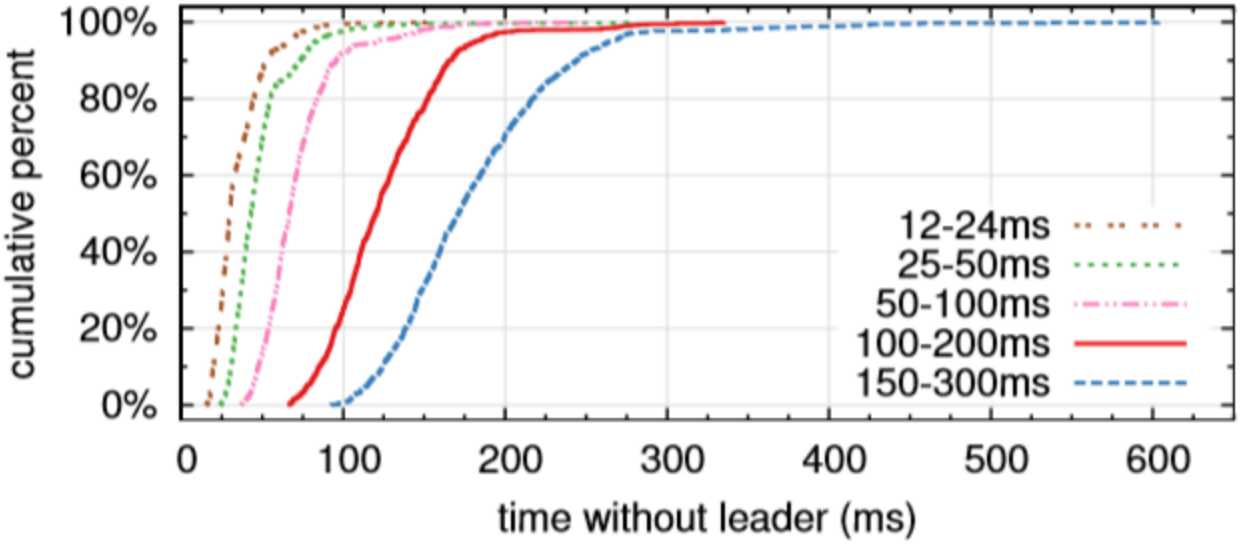


Survey Results



Result: Performance

- **The time to detect and replace a crashed leader.**
- **“x-y ms” means election timeout.**
  - Downtime can be reduced by reducing the election timeout.



- **Recommendation**
  - Use a conservative election timeout such as 150–300ms
  - Avoid unnecessary leader changes

References

- BOLOSKY, W. J., BRADSHAW, D., HAAGENS, R. B., KUSTERS, N. P., AND LI, P. Paxos replicated state machines as the basis of a high-performance data store. In Proc. NSDI'11, USENIX Conference on Networked Systems Design and Implementation (2011), USENIX, pp. 141-154.
- BURROWS, M. The Chubby lock service for loosely-coupled distributed systems. In Proc. OSDI'06, Symposium on Operating Systems Design and Implementation (2006), USENIX, pp. 335-350.
- CAMARGOS, L. J., SCHMIDT, R. M., AND PEDONE, F. Multicoordinated Paxos. In Proc. PODC'07, ACM Symposium on Principles of Distributed Computing (2007), ACM, pp. 316-317.
- CHANDRA, T. D., GRIESEMER, R., AND REDSTONE, J. Paxos made live: an engineering perspective. In Proc. PODC'07, ACM Symposium on Principles of Distributed Computing (2007), ACM, pp. 398-407.
- LAMPORT, L. Time, clocks, and the ordering of events in a distributed system. Commununications of the ACM 21, 7 (July 1978), 558-565.
- LAMPORT, L. The part-time parliament. ACM Transactions on Computer Systems 16, 2 (May 1998), 133-169.
- LAMPORT, L. Paxos made simple. ACM SIGACT News 32, 4 (Dec. 2001), 18-25.
- LAMPORT, L. Specifying Systems, The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley, 2002.
- LAMPORT, L. Generalized consensus and Paxos. Tech. Rep. MSR-TR-2005-33, Microsoft Research, 2005.
- LAMPORT, L. Fast paxos. Distributed Computing 19, 2 (2006), 79-103.
- LAMPSON, B. W. How to build a highly available system using consensus. In Distributed Algorithms, O. Baboaglu and K. Marzullo, Eds. Springer-Verlag, 1996, pp. 1-17.
- …

Questions