

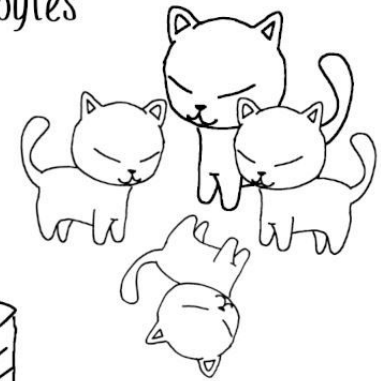
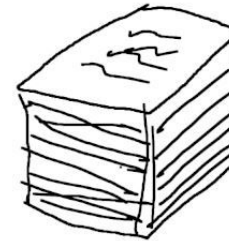
This zine was created for the
undergraduate distributed
systems course at UCSC.

Allen Aboytes, 2019

Why We Make Copies

A Note on Replication

By Allen Aboytes



Where to learn more?

Kenneth P. Birman. 2005. *Reliable Distributed Systems: Technologies, Web Services, and Applications*. Springer.

Andrew S. Tanenbaum and Maarten van Steen. 2006.

Distributed Systems: Principles and Paradigms (2nd Edition)

Mikito Takada. *Distributed systems for fun and profit*.

Robert Van Renesse and Fred B. Schneider. "Chain Replication for Supporting High Throughput and Availability."

UCSC's CMPS 128 Distributed Systems course

Who made this?

Hello everyone! My name is Allen,



I am an undergraduate Computer Engineering Student at the University of California, Santa Cruz.

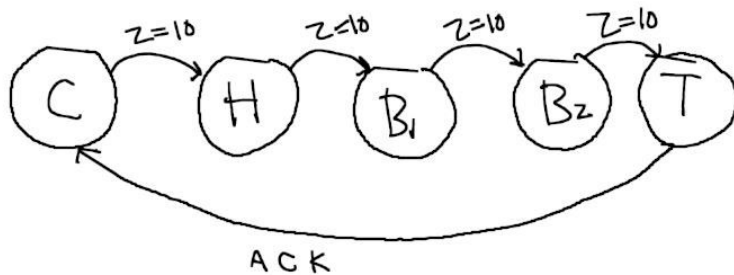


If anyone wants to chat about Distributed Systems, Security, or Networking:

aboytes@ucsc.edu

Chain Replication

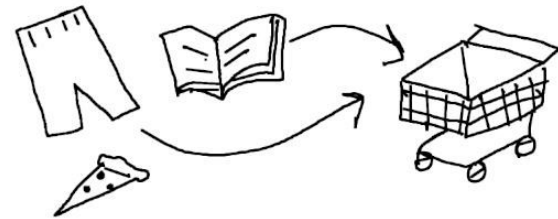
In this configuration nodes are organized in a list with both a head and a tail.



Writes are handled by the head and reads are handled by the tail. Writes are sent from the head to the tail, and then the client is sent an ACK once the tail responds.

Distributed Systems are Everywhere!!!

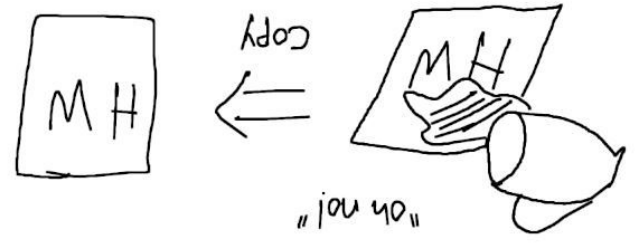
How do you think your Amazon shopping cart works? In the backend somewhere machines are keeping track of things.



A Distributed System is a group of machines working together to achieve a common goal. Like managing consistency in your shopping cart as you add and remove items.

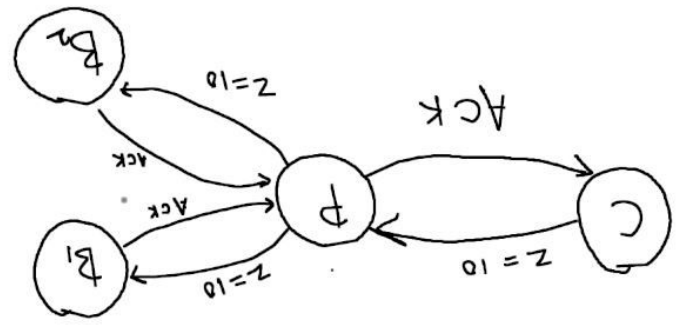
So Replication...

Replication is making copies of data. Machines communicate to share their state which is just an ordered sequence of events.



In a distributed system, any machine can fail. To counteract this a distributed system must provide fault tolerance. An that's where replication comes in.

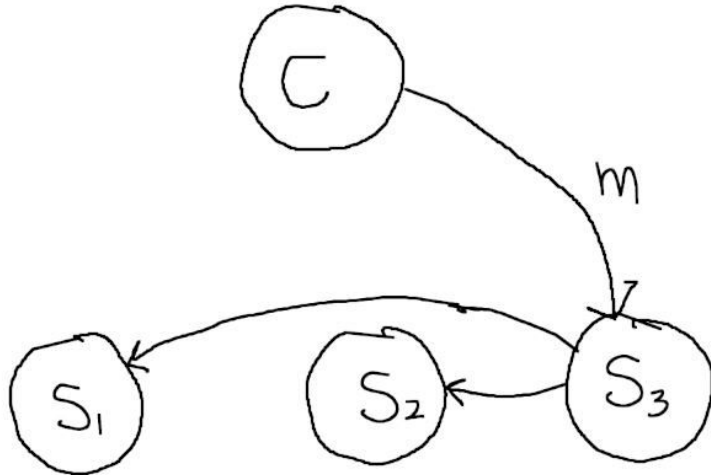
Primary Backup Replication



In PB replication the only one leader node handles all requests. Reads happen at the primary and writes get broadcasted to backups.

Passive Replication

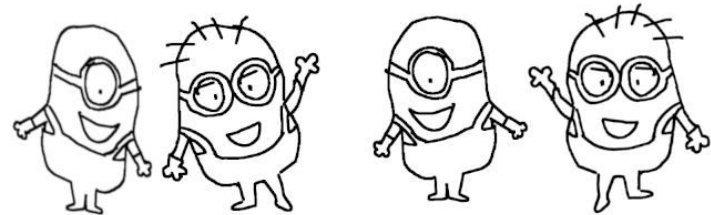
In this mode of operation one dedicated leader node handles all requests and executes and updates other nodes of the state



The More the Merrier

Replication is useful for many reasons, it provides:

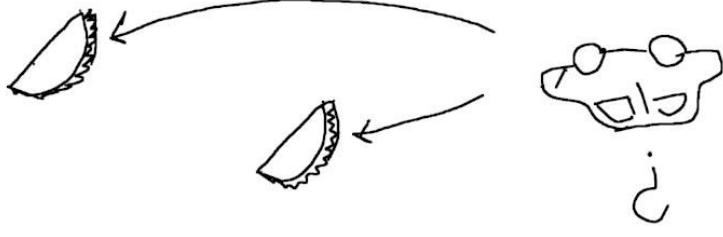
- fault-tolerance
- horizontal scaline
- locality



We wouldn't want to loose important numbers like the balance in our bank account. Fault-tolerance ensures we can handle failures of machines working together.

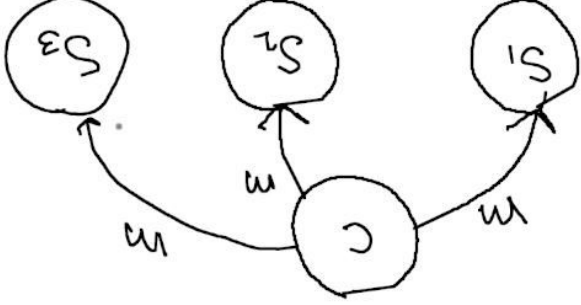
"Real quick, what's Horizontal Scaling and Locality?"

Horizontal Scaling is adding more machines to increase the amount of messages that can be handled. And locality is the concept of having resources physically closer together. Imagine having many taco bell restaurants in your city and finding it easier and cheaper to go to the closest one.



Active Replication

Also known as State Machine Replication, the idea originally introduced by Leslie Lamport says that each client request is processed by all servers.



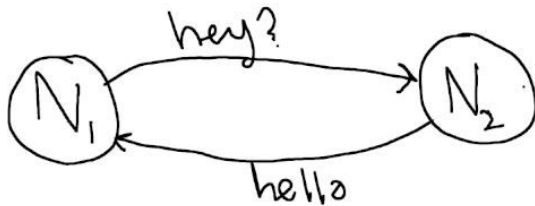
In order to achieve this each process run by a server node must be deterministic so that each state if executed in the same order is the same.

Types of Replication

There are a lot of ways one can achieve a replicated state across nodes in a distributed network. We will talk about two kinds:

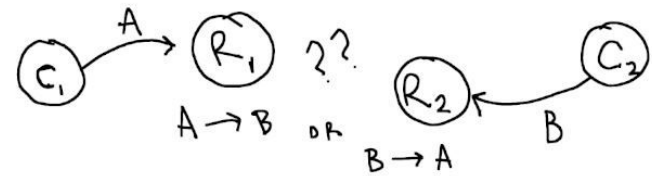
- Primary Backup Replication
- Chain Replication

Before we get to that we will go over active and passive replication which describe how nodes behave during a procedure.

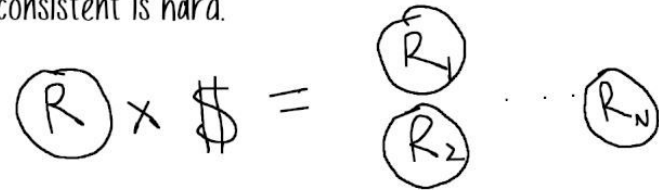


The Horrors of Replication

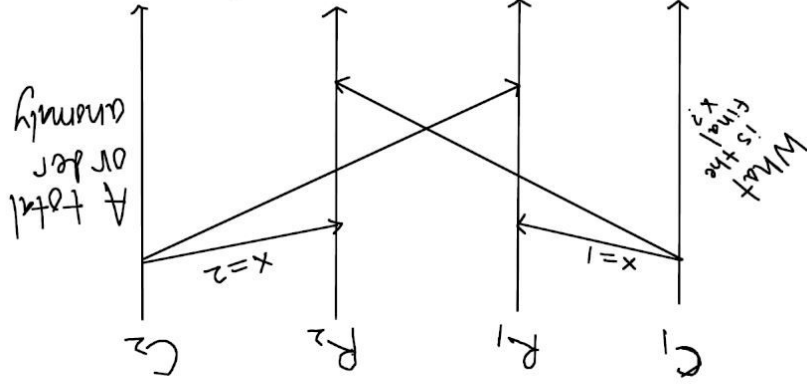
Replication can be challenging.



Having many resources can be expensive and keeping each replica consistent is hard.



Consistently a Failure

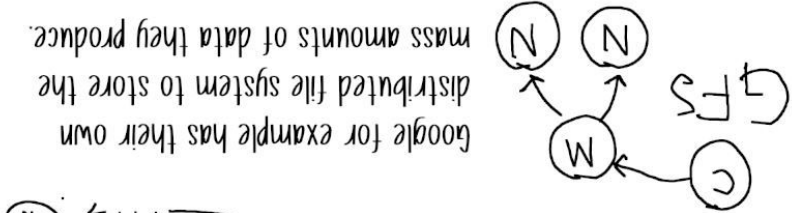


Issues can arise when events in a distributed system occur. Order really matters. Messages may arrive out of order between replicas.

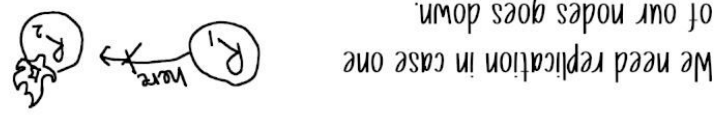
Replication IRL



Replication can be used for distributed data stores.



Google for example has their own distributed file system to store the mass amounts of data they produce.



We need replication in case one of our nodes goes down.