Distributed Systems

An Intro

TYR

Definition

- A collection of computers running in a network
- communicate and coordinate towards a goal

Things to consider

- Communication / coordination
- fault tolerance (software crash, hardware failure, network down)
- Scalability

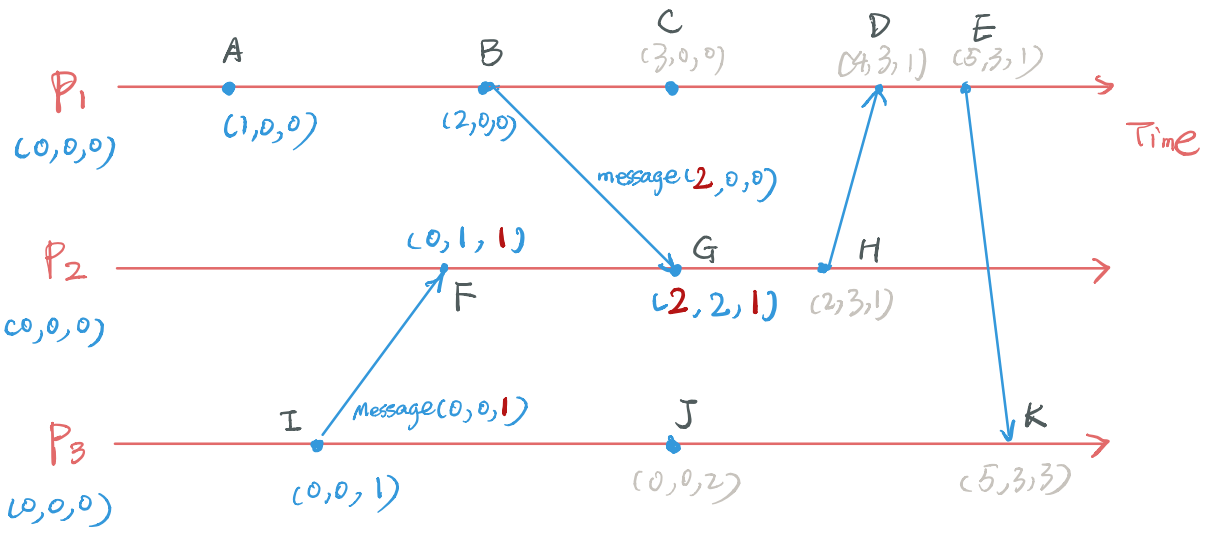
Concepts

- Node : an independent computer
- Network : the channel for communication
  - could be heterogeneous
  - communication has a cost!
- Protocol : rules for transmitting information

Comparison

	Solo	Cloud computing	fully distributed
global clock	strict	yes	no
environment	simple, controlled	managed, a little moving parts	nodes come and go
admin	single	single	no
security	easy to implement	harder	hardest
fault-tolerance	bad	good	best
protocol	n/a	private/easy to change	public/hard to change
peer discovery	n/a	static/dynamic (consul)	gossip/
upgrade	easy	rolling / blue-green	hard (consensus)
consensus	n/a	static/paxos/zab/raft	PBFT/PoX

Vector Timestamp



Causally-related

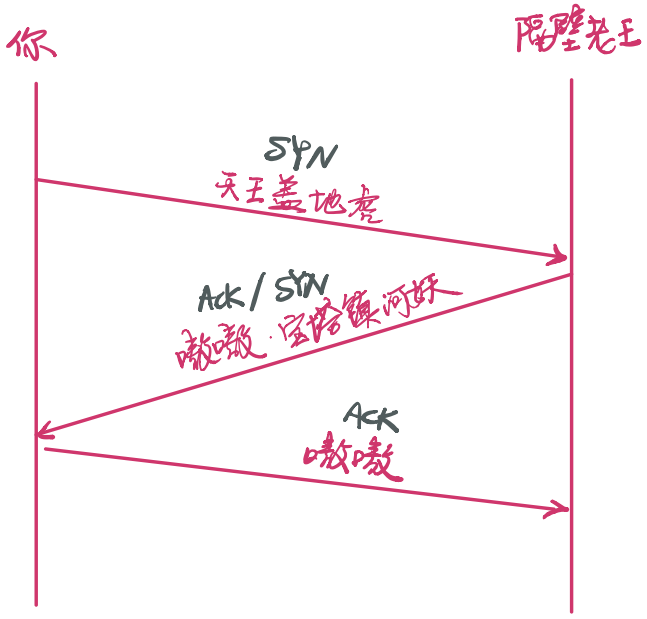
A → B (1,0,0) < (2,0,0)

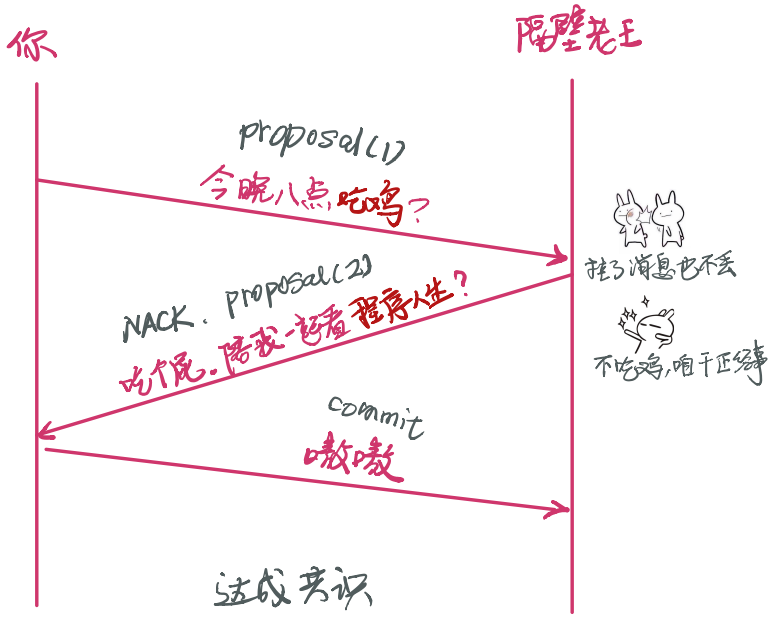
B → G (2,0,0) < (2,2,1)

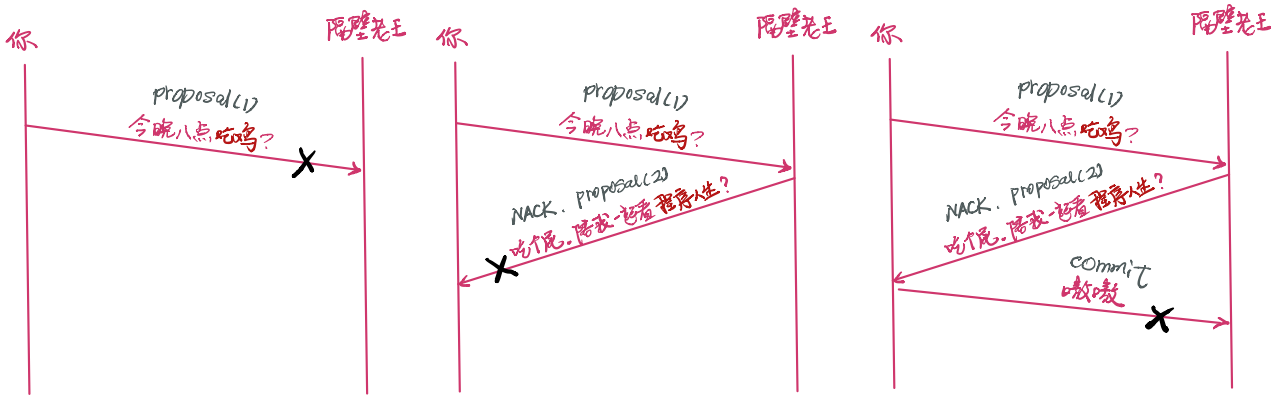
Concurrent

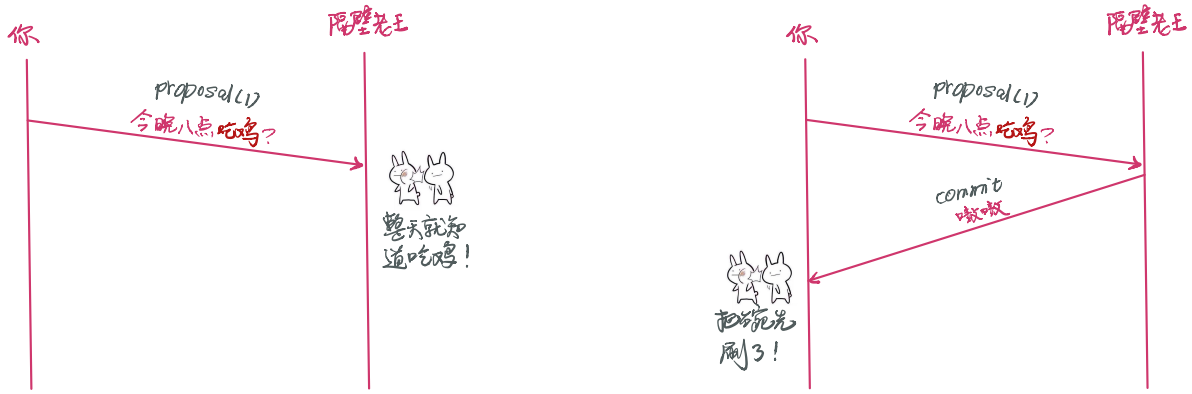
C & G (3,0,0) ||| (2,2,1)

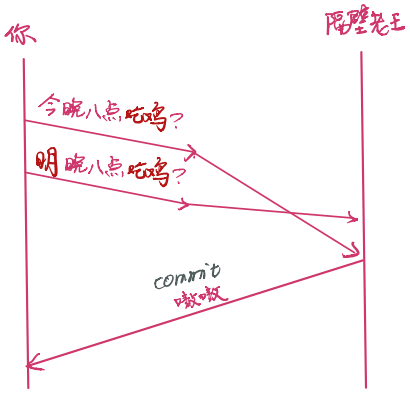
I & C (0,0,1) ||| (3,0,0)

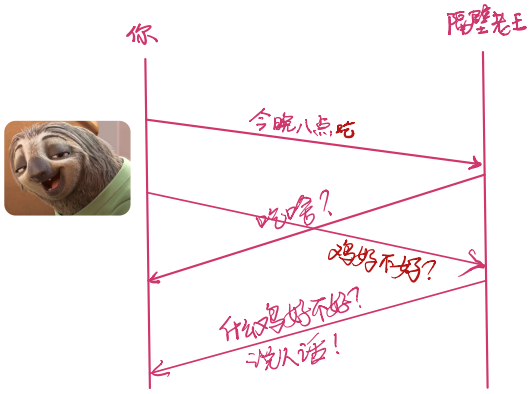



你
陪跑老王
proposal(1)
今晚八点吃鸡?
挂了消息也不丢
NACK. proposal(2)
吃个屁. 陪我一起看程序人生?
不吃鸡,咱干正经事
commit
嗷嗷
达成共识


你
陪跑老王
proposal(1)
今晚八点吃鸡?
你
陪跑老王
proposal(1)
今晚八点吃鸡?
NACK. proposal(2)
吃个屁. 陪我一起看程序人生?
你
陪跑老王
proposal(1)
今晚八点吃鸡?
NACK. proposal(2)
吃个屁. 陪我一起看程序人生?
commit
嗷嗷

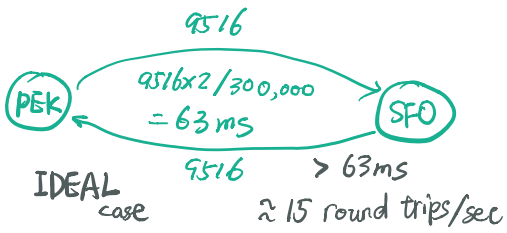

你
陪跑老王
proposal(1)
今晚八点吃鸡?
整天就知道吃鸡!
你
陪跑老王
proposal(1)
今晚八点吃鸡?
commit
嗷嗷
把你完先刷3!

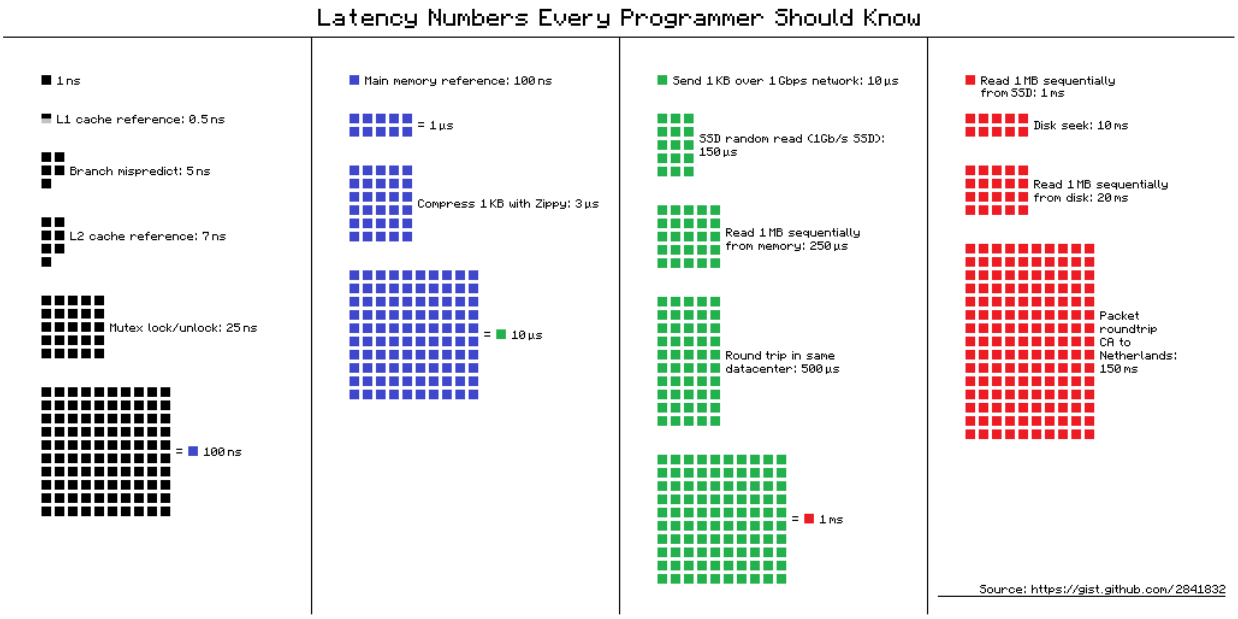

你
隔壁老王
今晚八点吃鸡?
明晚八点吃鸡?
commit
嗷嗷叫


你
隔壁老王
今晚八点吃
吃鸡?
鸡好不好?
什么鸡好不好?
没人话!

Prejudice

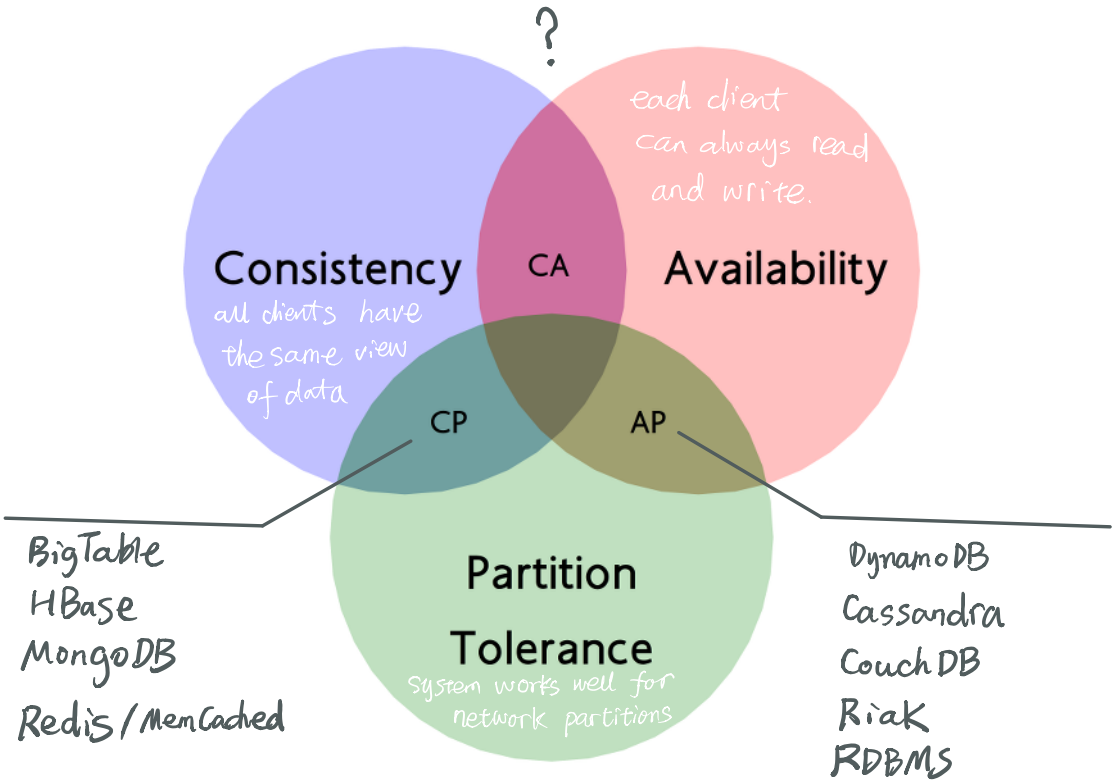
- Network is reliable
  - Can we rely on network to deliver the message?
- Network is homogeneous
  - wired, wireless, cellular, satellite
  - speed varies a lot
- Latency is neglectable
  - L = distance / C





- Bandwidth is not a problem
  - upload speed is pretty limited for home users
- Network is secure
- Topology won't change
  - Node could come and go

CAP theorem

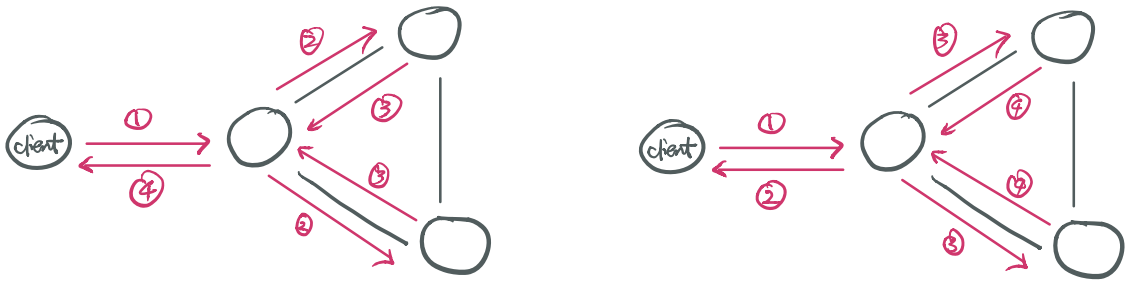


Consistency Models

- linearizability
- Sequential consistency
- Causal consistency
- Eventual strong consistency
- Eventual consistency
- Monotonic read consistency
- Monotonic write consistency
- Read-your-writes consistency
- Writes follows reads consistency
- Serializability

Partition and Replication

- Split data to multiple nodes
- Copy data over to ensure consistency
  - sync : consisteney over performance
  - async : performance over consistency



Algorithms

- Gossip
- Consistent hashing