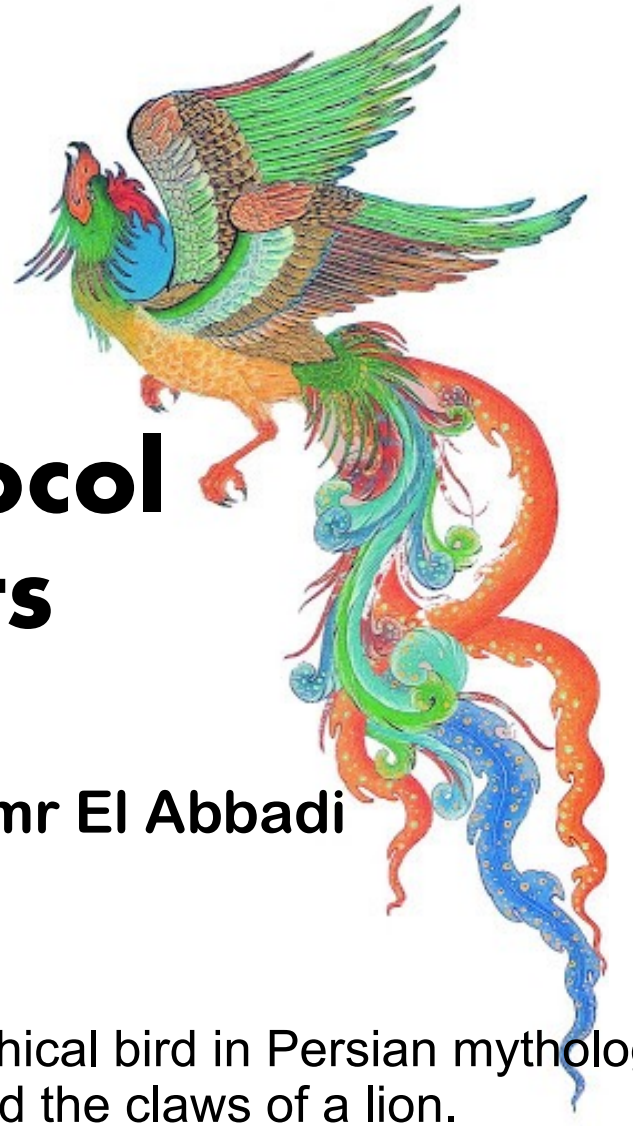
UC Santa Barbara
Computer Science Department

SeeMoRe: A Fault-Tolerant Protocol for Hybrid Cloud Environments

Mohammad Javad Amiri, Sujaya Maiyya, Divyakant Agrawal, Amr El Abbadi



ICDE 2020 International Conference on Data Engineering

SeeMoRe is derived from Seemorq, a benevolent, mythical bird in Persian mythology which appears as a peacock with the head of a dog and the claws of a lion.

Fault Tolerance

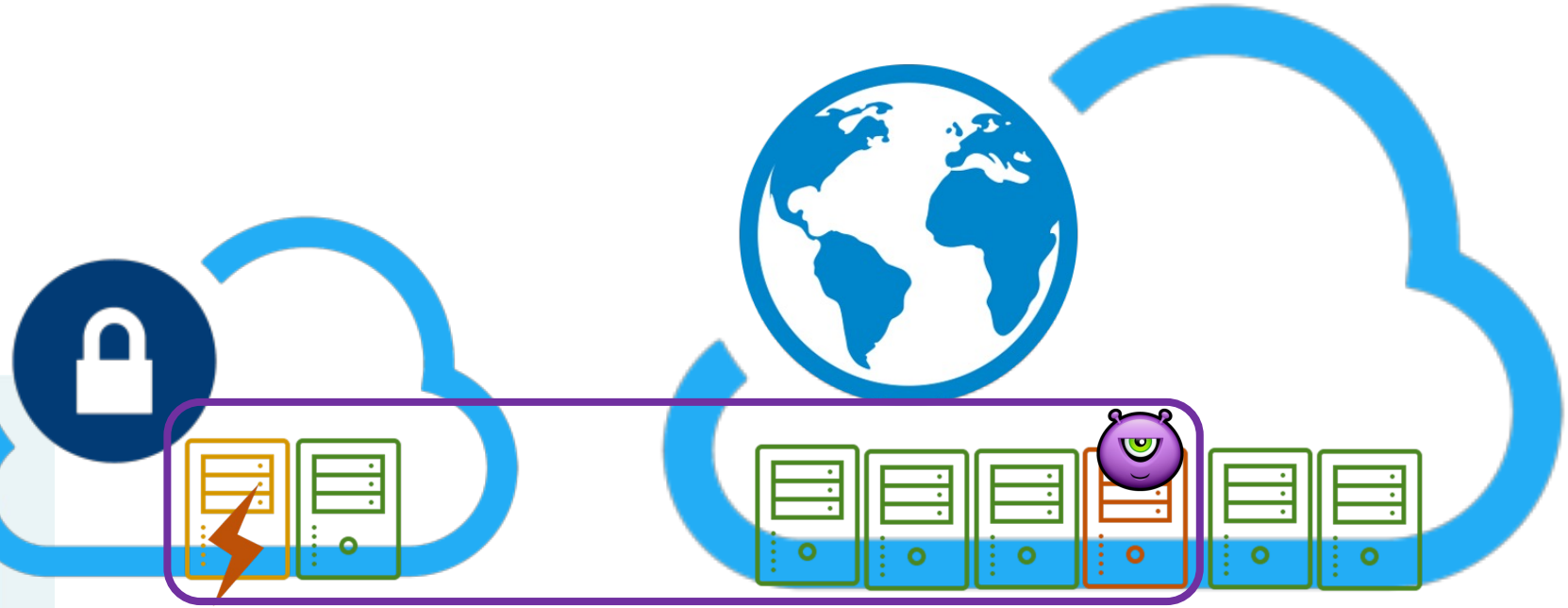
- Build systems that tolerate machine and network faults
- Replicate data on multiple servers to enhance availability
  - Uses State Machine Replication: All servers execute same commands in same order
  - Needs consensus among different servers

Large Enterprises

- Have their own Geo-replicated fault-tolerant clouds

Small Enterprises

Lack of resources to guarantee fault tolerance



Nodes in the private cloud are trusted (crash-only)

Nodes in the public cloud are untrusted (Byzantine)

Can we benefit from both worlds?

Consensus Problem

A set of distributed nodes need to reach agreement on a single value

Types of systems: synchronous and asynchronous

Types of failure: Crash, e.g., Paxos, and Byzantine, e.g., PBFT

(Multi-)Paxos

Proposal

Primary to backups

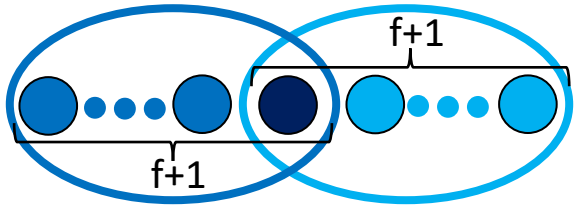
Decision Making

Backups to Primary

Centralized | O(n)

At Most f Crash Failures

quorum A quorum B



Network: 2f+1
Quorum: f+1
Intersection: 1

Phases: Two
Messages: O(n)
Quorum: f+1

Practical Byzantine Fault Tolerance

Proposal

Primary to backups

Proposal Validation

Backups to All

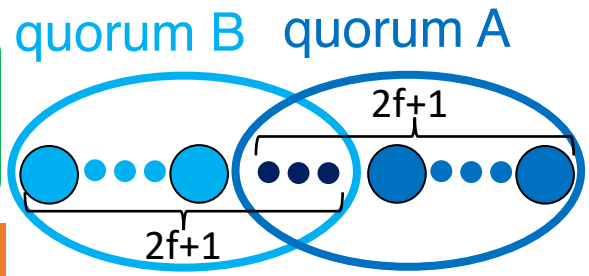
Decentralized $O(n^2)$

Decision Making

All to All

Decentralized $O(n^2)$

At Most f Malicious Failures

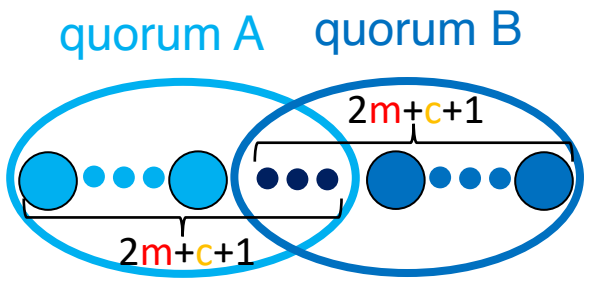


Network: 3f+1
Quorum: 2f+1
Intersection: f+1

Phases: Three
Messages: $O(n^2)$
Quorum: 2f+1

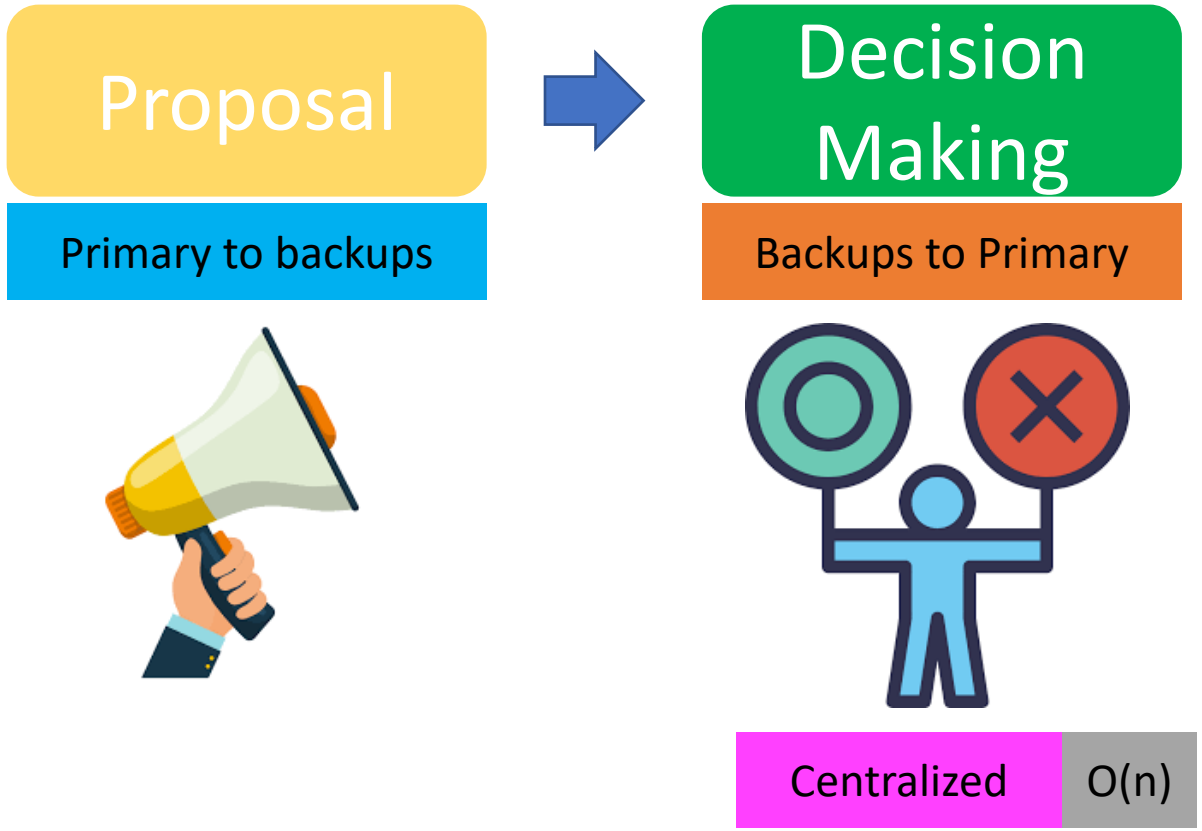
SeeMoRe Model

- Tolerate at most m Malicious and at most c crash faults
  - f = m + c
  - Quorum: 2m + c + 1
  - Intersection: m + 1
  - Network: 3m + 2c + 1



Mode 1: Trusted Primary, Centralized Coordination

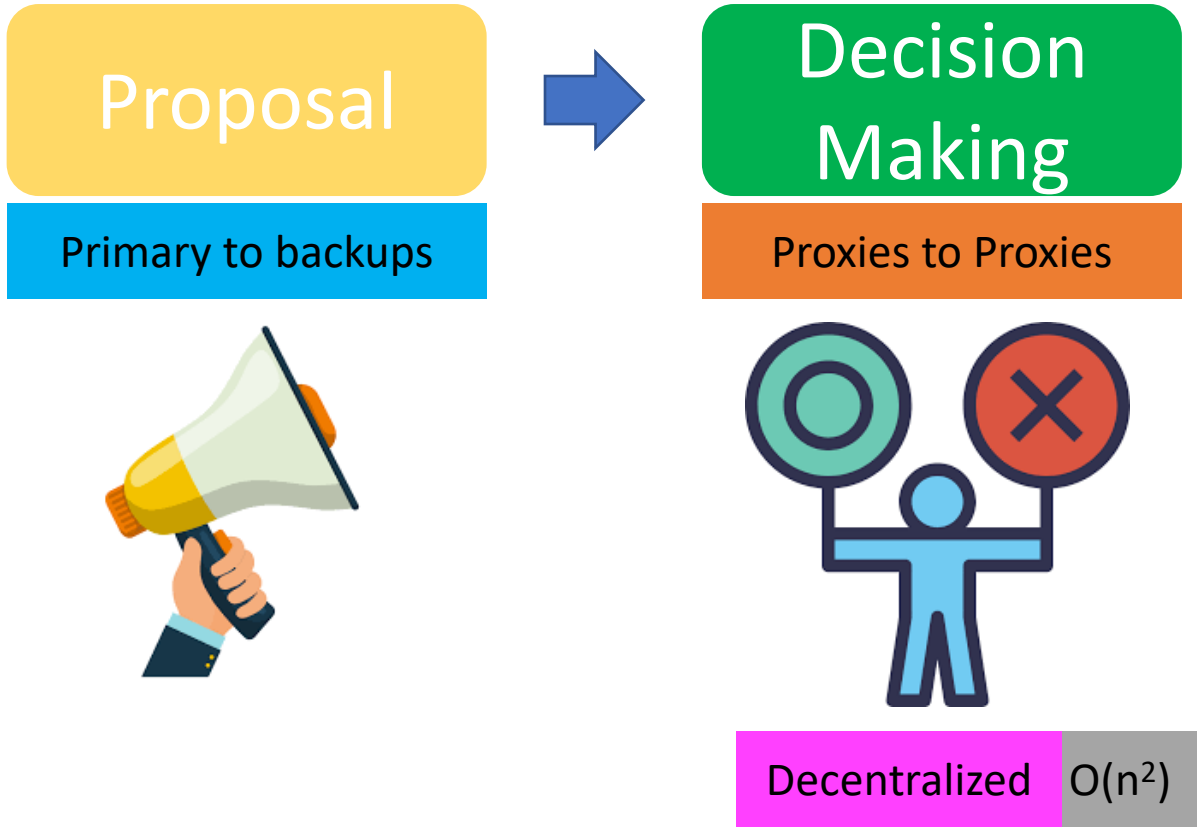
- The primary is in the private cloud (Trusted)
- Backups are in both private and public cloud



Phases: Two
Messages: O(n)
Quorum: 2m+c+1

Mode 2: Trusted Primary, Decentralized Coordination

- The primary is still in the private cloud (Trusted)
- The private cloud is not involved in the second phase
- Proxy nodes: *3m+1* nodes from the public cloud



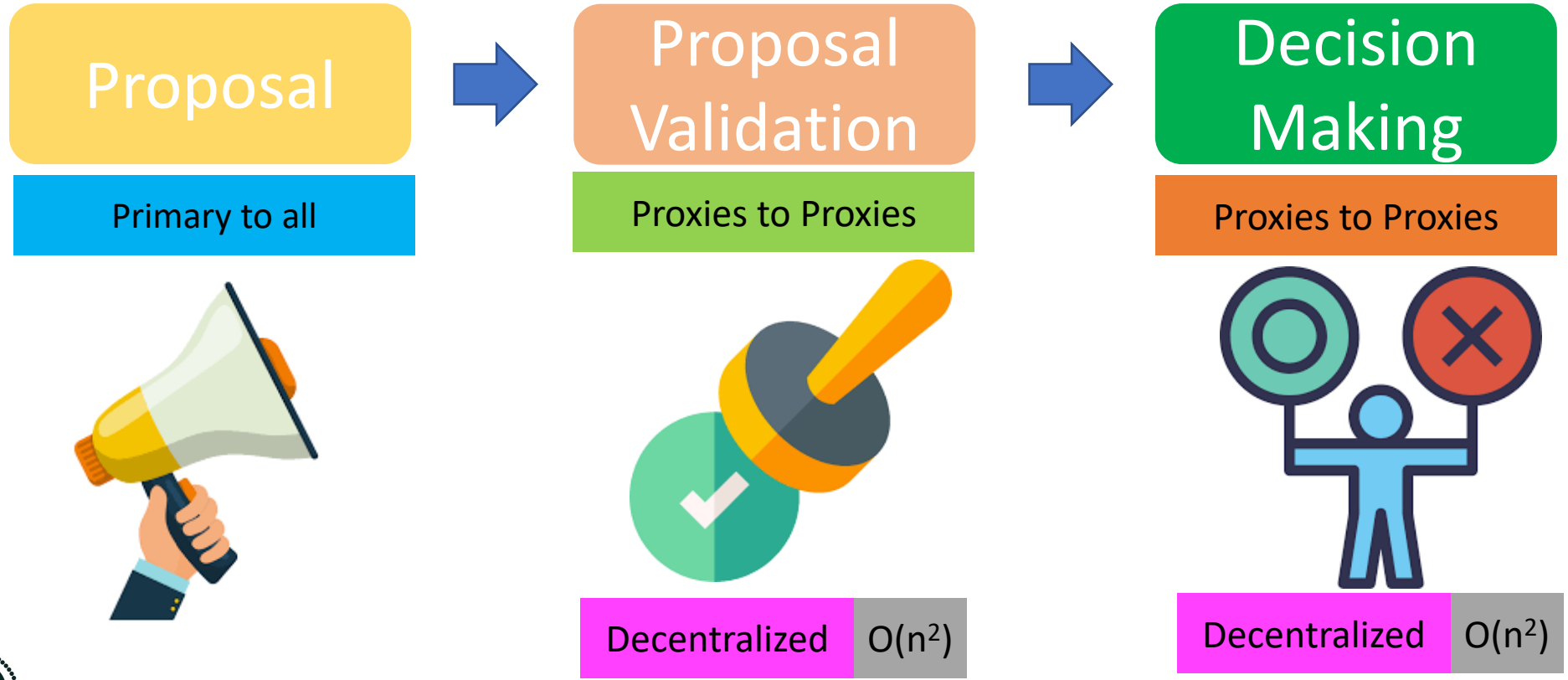
Goal:
Reduce the load on the private cloud

Phases: Two
Messages: $O(n^2)$
Quorum: 2m+1

Mode 3: Untrusted Primary, Decentralized Coordination

- The primary is in the public cloud (Untrusted)
- The private cloud is not involved in any phases
- Proxy nodes: *3m+1* nodes from the public cloud

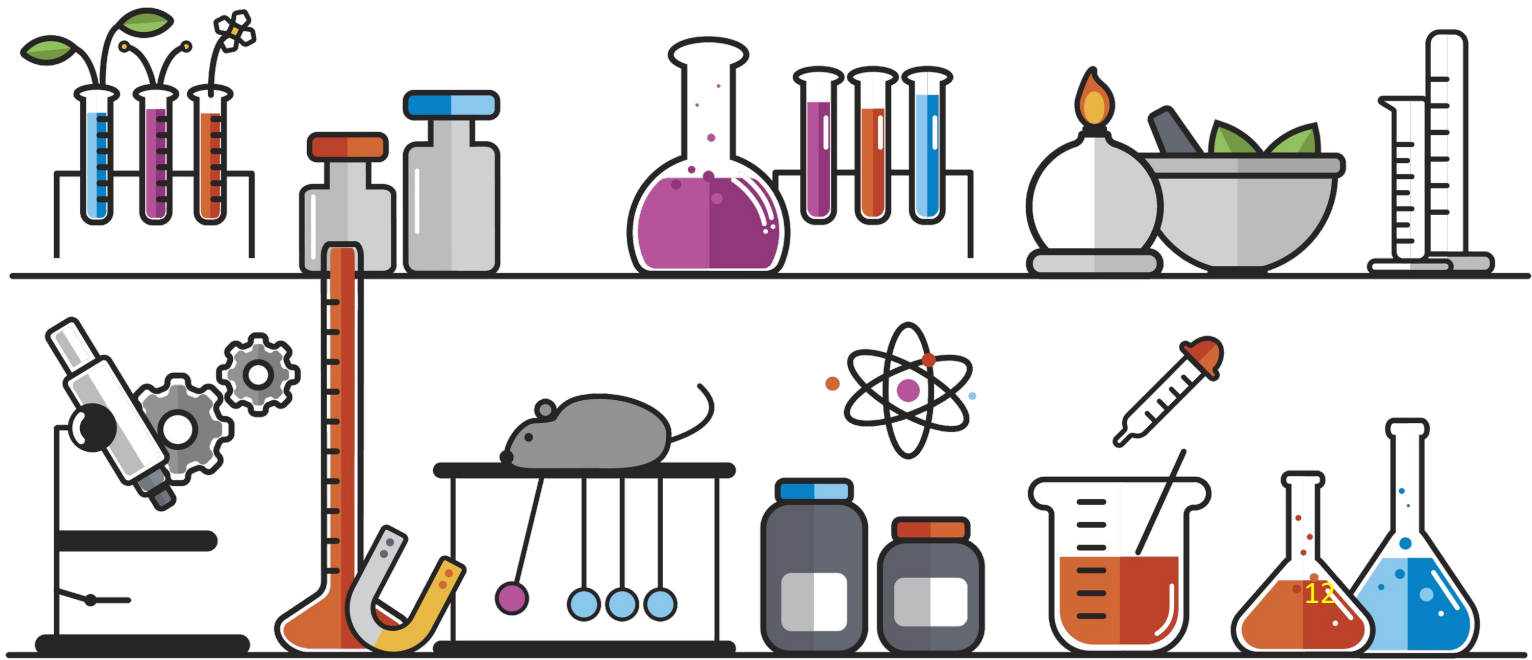
Goal:
Reduce latency when there is a large network distance between clouds



Phases: Three
Messages: $O(n^2)$
Quorum: 2m+1

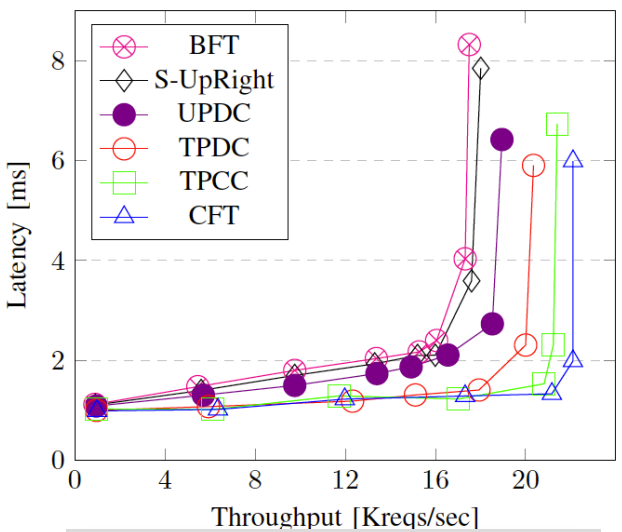
Experimental Settings

- Systems:
  - Crash Fault-Tolerant: Paxos
  - Byzantine Fault-Tolerant: PBFT
  - Hybrid Fault-tolerant: UpRight
  - SeeMoRe
    - TPCC
    - TPDC
    - UPDC
- Platform: Amazon EC2
- Measuring performance
  - Throughput
  - Latency



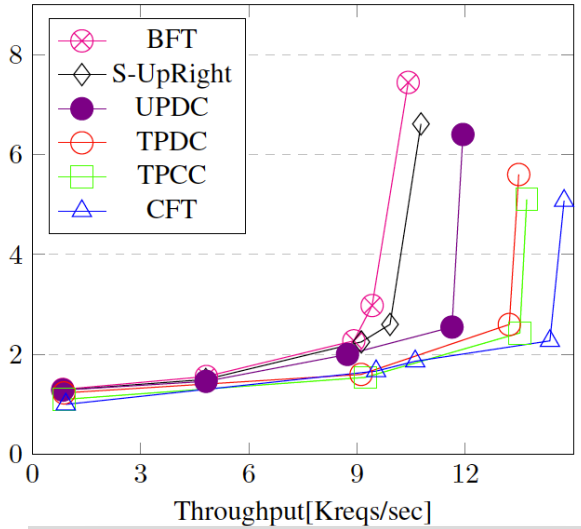
Fault Tolerance Scalability

f = 2 (c = 1, m = 1)



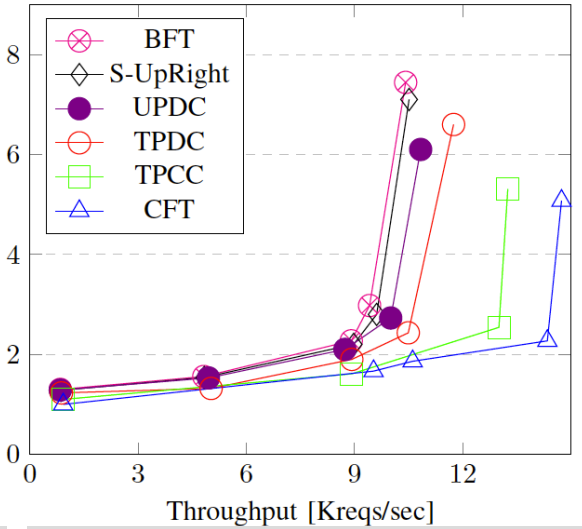
CFT:5, PFT:7, SeeMoRe, UpRight:6

f = 4 (c = 2, m = 2)



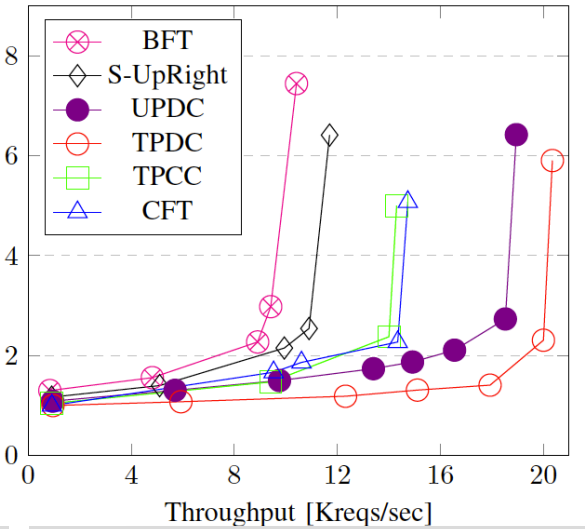
CFT:9, PFT:13, SeeMoRe, UpRight:11

f = 4 (c = 1, m = 3)



CFT:9, PFT:13, SeeMoRe, UpRight:12

f = 4 (c = 3, m = 1)



CFT:9, PFT:13, SeeMoRe, UpRight:10

The performance of the TPCC mode becomes very close to CFT (8% difference in their peak throughput).

TPCC are TPDC show similar performance: the trade-off between the quorum size and the message complexity

By increasing m, the network size of SeeMoRe becomes closer to the BFT network size

TPDC mode processes a request in the public cloud which needs only 4 replicas while TPCC requires 10 replicas

Scalability Across Multiple Data centers

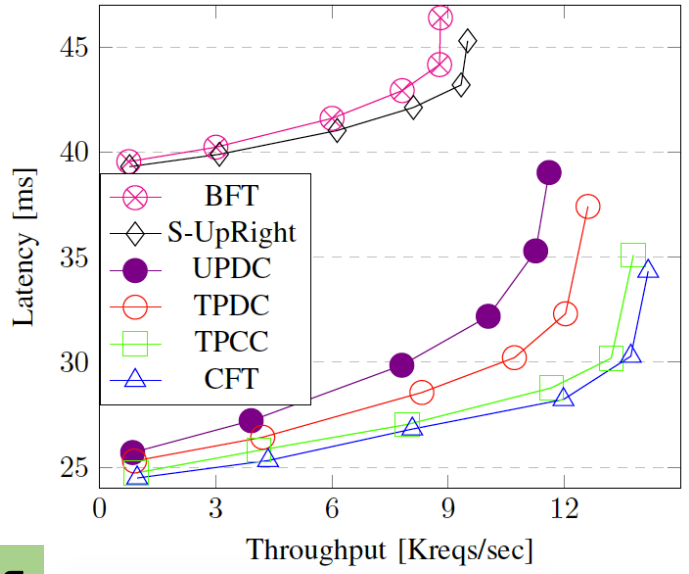
c =1, m =1

0/0 Benchmark

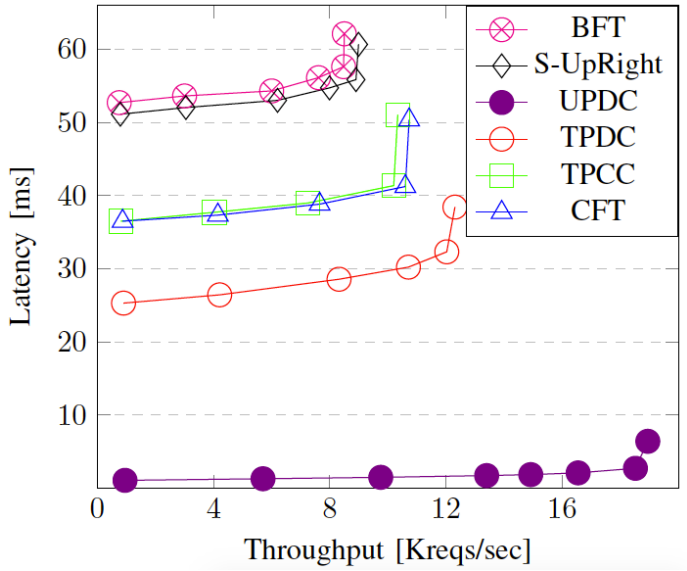
RTT: 22ms

- Private Cloud: California
- Public Cloud: Oregon

Clients close to the Private Cloud



Clients close to the Public Cloud



Cross-cloud communication

SeeMoRe, CFT	2 phases
BFT, S-UpRight	3 phases

UPDC	0 phases
TPDC	2 phases
TPCC, CFT	3 phases
BFT, S-UPRight	4 phases

Conclusion

- SeeMoRe, a hybrid State Machine Replication protocol to tolerate both crash and malicious failures in a public/private cloud environment
- Distinguishes between crash failures (occurs within the trusted Private cloud) and malicious failures (occur in the public cloud)
- To be used by small enterprises that own a small set of servers and intend to rent servers from public cloud providers.
- Can execute in any one of three modes, TPCC, TPDC, and UPDC, And can dynamically switch among these modes.
- Future work: SeeMoRe can be used in the context of permissioned blockchain systems.


THANK YOU!
Questions?