Channel Into Universe Of Eventually Perfect Distributed Systems
Fundamental techniques and building blocks
Are fundamentals still important?
Your System

Your Trade-Offs

@lenadro
@lenadro

hm...

Theory

Gap

Practice

@lenadro
There are challenges
Road to Correctness and Understanding

@lenadroid
Simple problems become hard
Ordering is Hard
Lamport Clock

2 and 3 could be concurrent!
Vector Clock

{A:1, B:1} {A:2, B:1} {A:3, B:1, X} {A:4, B:1, Y}

{B:1} {A:2, B:1} {A:2, B:1, C:1} {A:2, B:1, C:2}
Agreement In Distributed Systems
Two Phase Commit
Blocking Failure in Two Phase Commit

Nodes are blocked! Can’t decide!
Hm... it's blocking when there're

Two Phase

Three Phase
Hm... it's blocking when there are Two Phase

Three Phase

Might be inconsistent in asynchronous environment
“Distributed consensus is impossible in asynchronous system where at least one node can fail.”
Hm… it’s blocking when there’re
Three Phase
Might be inconsistent in asynchrono
FLP
Two Phase

Paxos
Classical
Fast Paxos
Multi-Paxos
Vertical
Cheap Paxos
Zab
Raf
Chandra-Toueg

@lenadro
Paxos
Trade-offs

Weak or strong leader?

Quorum size?

Number of Failures Tolerated?

Optimizations

Proposal Copying

Distinguished Proposer

Combining Roles

Strategies for Proposal Numbers
Discovering New Trade-offs and Optimizations

Quorum intersection revised

Quorum based value selection

Proposal numbers uniqueness

And many more…

cl.cam.ac.uk/techreports/UCAM-CL-TR-935.pdf by Dr. Heidi Howard
Consistent Replication?
Amazon Aurora: On Avoiding Distributed Consensus for I/Os, Commits, and Membership Changes

Figure 3: Storage Consistency Points

dl.acm.org/citation.cfm?id=3183713.3196937
Conflict-Free Replicated Data Types
add X
delete X
delete X
add X
Cosmos DB

x: \{ b, c \}

x: \{ a \}

x: \{ a, b, c \}
Completeness: can all nodes discover all the failures?

Accuracy: how precise can a node be in its failure suspicions?
Understanding Trade-offs Helps

✓ To make the right choices
✓ To know what correct means for us
✓ To verify and maintain correctness in practice
Maintaining Correctness In Real Systems

@lenadro
Model Checking
Verifying and Maintaining Correctness in Practical Real-World Systems

@lenadro
Kafka System Tests

SESSION REPORT (ALL TESTS)
ducktape version: 0.7.5
session_id: 2019-05-26--001
run time: 823 minutes 18.415 seconds
tests run: 466
passed: 387
failed: 10
ignored: 69

450+ system tests, 6800+ unit tests, 600+ integration tests

testing.confluent.io/confluent-kafka-system-test-results
confluent.io/blog/apache-kafka-tested
Cassandra Tests

Replay testing
Dynamic test generation
Property-based testing and fuzzing
Distributed tests and fault-injection
Upgrade testing

cassandra.apache.org/blog/2018/10/17/finding_bugs_with_property_based_testing.html
cassandra.apache.org/blog/2018/08/21/testing_apache_cassandra.html
+ Model-checking
+ Property-based testing and fuzzing
+ Performance and upgrade testing
+ Unit and integration testing
+ Fault injection
+ Attention to exception handling logic
+ Code reviews
Take-Aways
✓ Know your trade-offs
✓ Create understandable systems
✓ Invest in correctness, it doesn’t come for free
✓ Don’t trust: test and verify
✓ Automate, but be ready when things fail
✓ Remember the real problem you are solving
Signs you are in a bubble: you only talk to people in your bubble, you don’t trust going outside your bubble.

How do we better combine bubbles so we get more different points of view?
#SystemsYouUnderstand