EPaxos Revisited

Sarah Tollman, Seo Jin Park, and John Ousterhout

PLATFORMLAB
Stanford University

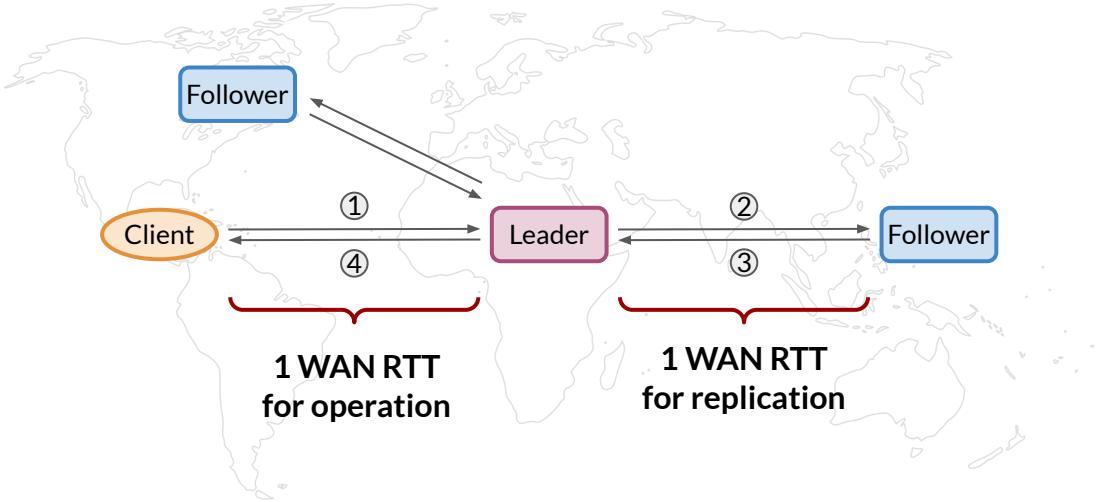
CSAIL

Overview

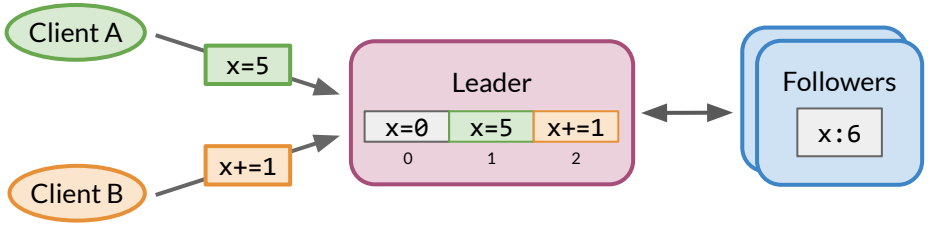
- **Problem**
  - Traditional WAN consensus is slow
  - Many *fast path* protocols do not work well in a WAN
  - Egalitarian Paxos (EPaxos), SOSP '13, claims to achieve low WAN latency
- **Contributions**
  - More comprehensive EPaxos evaluation
  - Timestamp-Ordered Queueing (TOQ)
    - Leverage synchronized clocks to reduce conflicts
- **Results**
  - Conflicts degrade EPaxos performance
    - Confirmed low median latency
    - P99 latency up to 4x worse than traditional Multi-Paxos
  - TOQ reduces conflict by 50%+
    - Decreases mean latency by up to 7.5%

Traditional Consensus (e.g. Multi-Paxos)

- 2 WAN RTTs

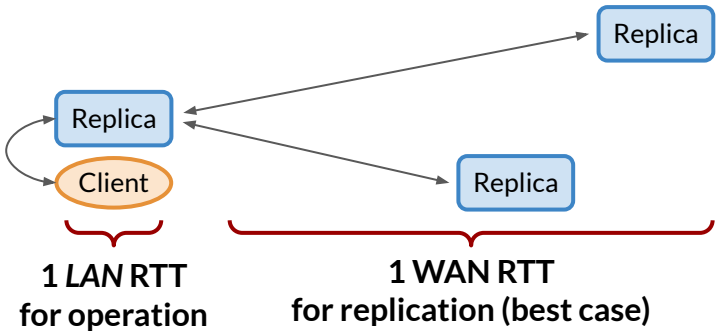


- Leader ensures consistency
  - Ordering replicated to quorum

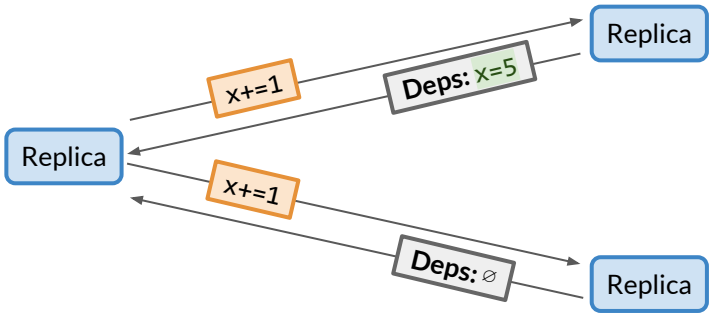


Egalitarian Paxos (EPaxos)

- Convert one WAN RTT to LAN
  - Leaderless
  - Clients issue operations to local replicas

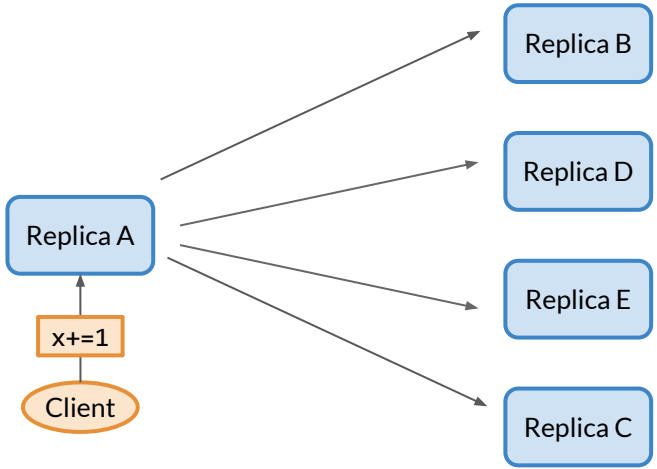


- Two purposes for WAN RTT
  1. Make operation durable
  2. Collect operation *dependencies*
     - Operations that are non-commutative with the current operation



Dependencies & Conflicts

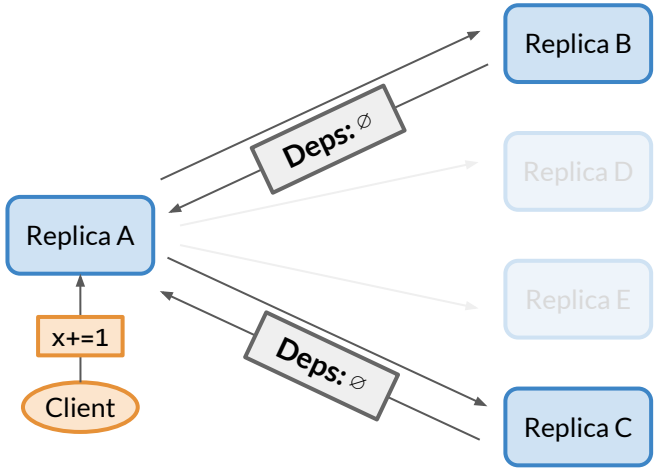
Fast path: 1 WAN RTT



Slow path: 2 WAN RTTs

Dependencies & Conflicts

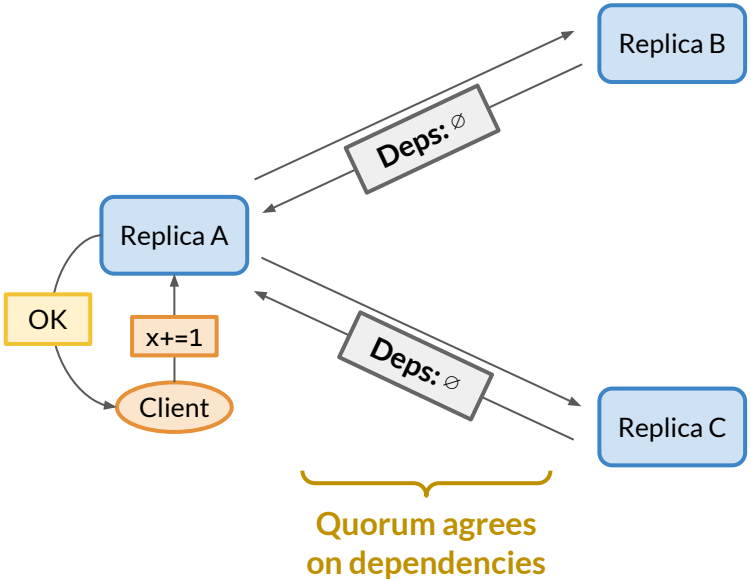
Fast path: 1 WAN RTT



Slow path: 2 WAN RTTs

Dependencies & Conflicts

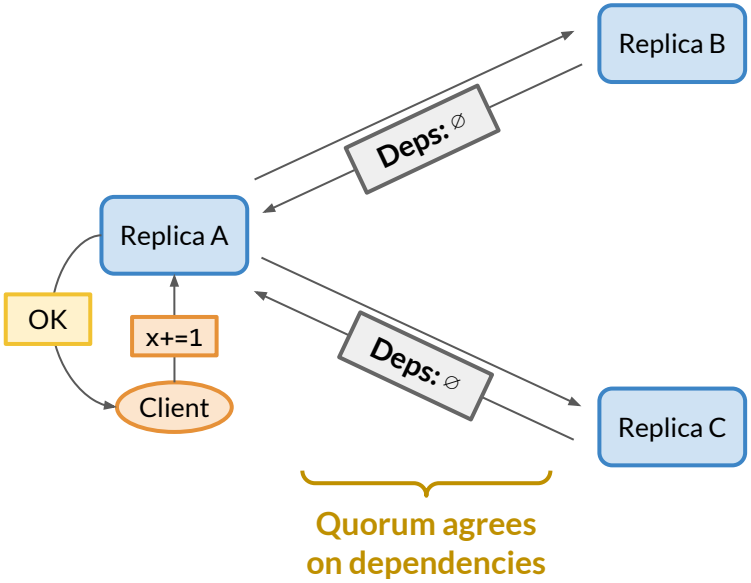
Fast path: 1 WAN RTT



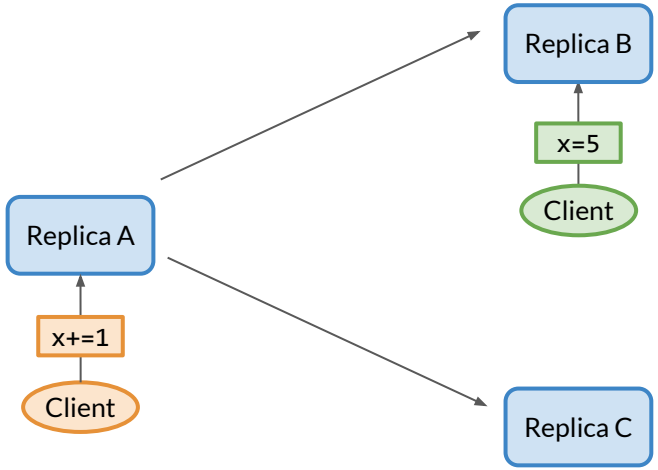
Slow path: 2 WAN RTTs

Dependencies & Conflicts

Fast path: 1 WAN RTT

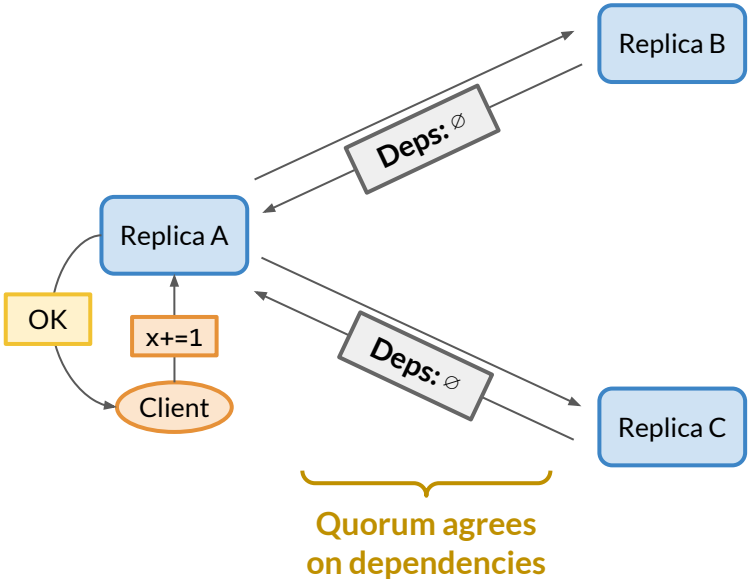


Slow path: 2 WAN RTTs

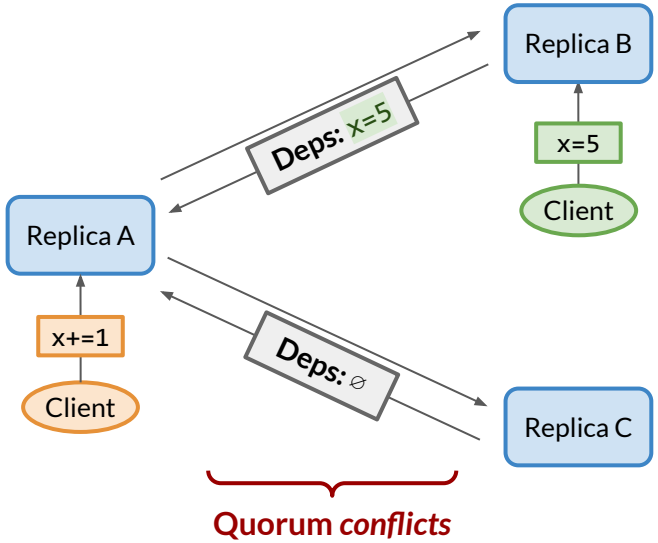


Dependencies & Conflicts

Fast path: 1 WAN RTT

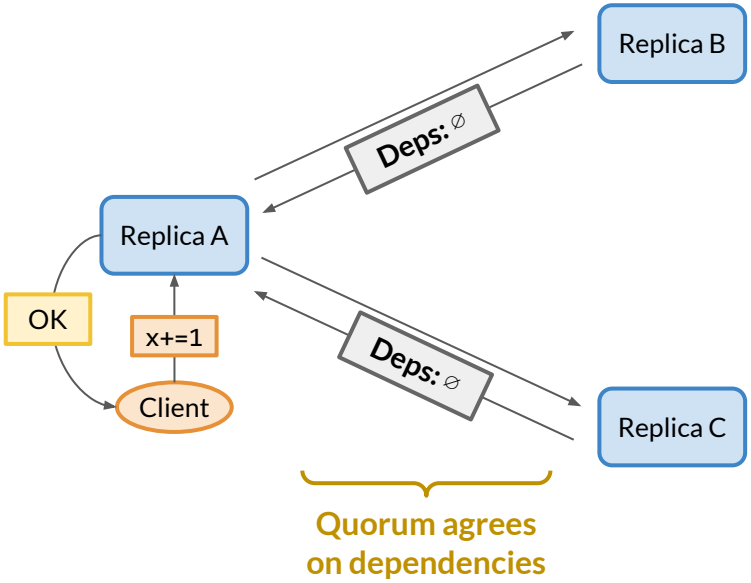


Slow path: 2 WAN RTTs

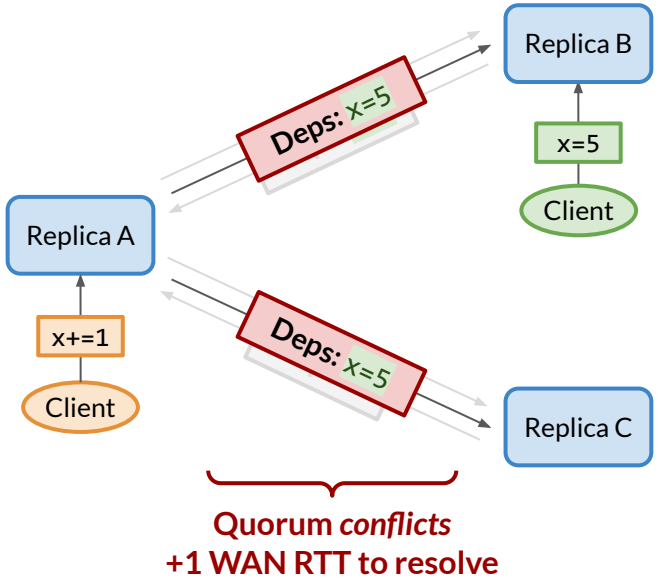


Dependencies & Conflicts

Fast path: 1 WAN RTT

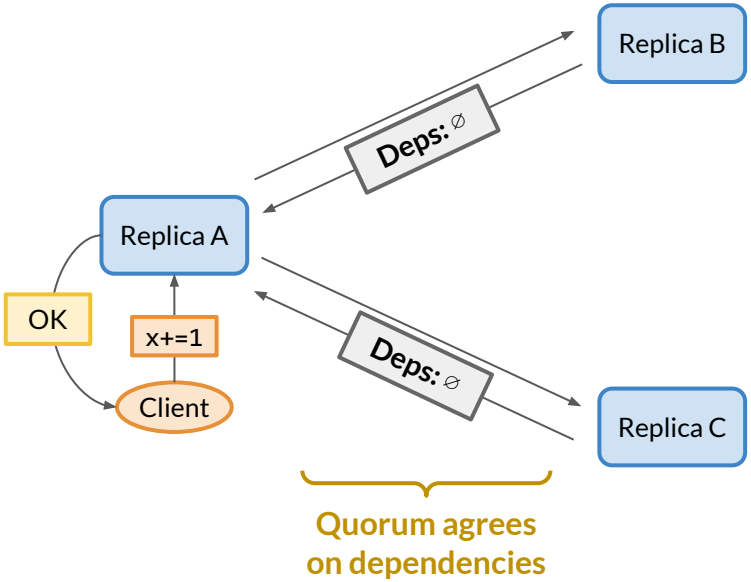


Slow path: 2 WAN RTTs

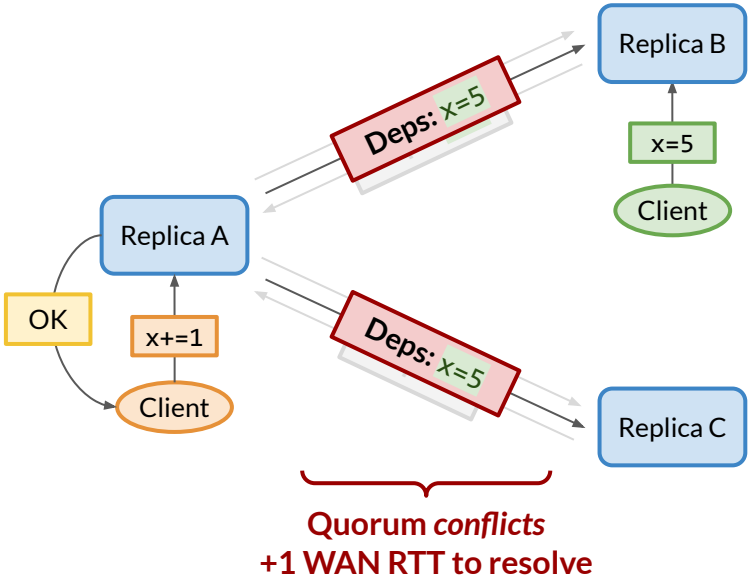


Dependencies & Conflicts

Fast path: 1 WAN RTT



Slow path: 2 WAN RTTs

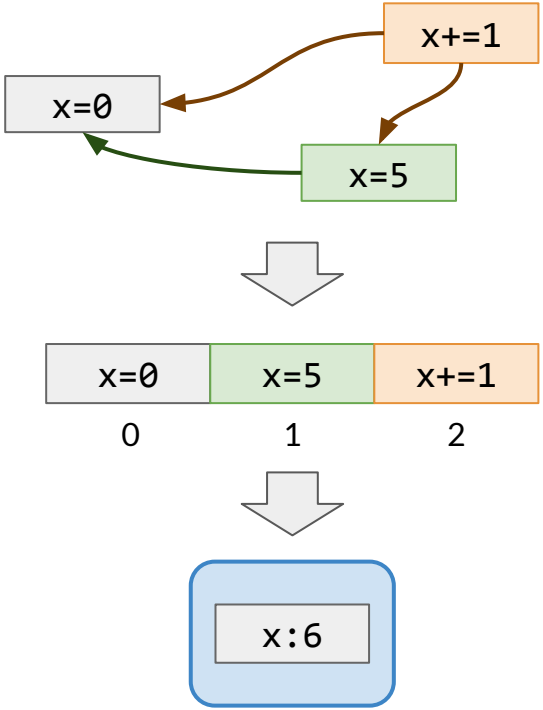


EPaxos Commit: Different From MPaxos

- Multi-Paxos
  - Durable
  - Fully ordered
  - Ready to execute
  - *Can determine result*
- EPaxos
  - Durable
  - Dependencies finalized
  - Will eventually execute
  - *May not know result or ordering*

Execution

1. Wait for notification of dependencies' commits
   - Execution latency ≥ commit latency
2. Determine ordering
   - Traverse dependency graph
   - Deterministic, replicas do this independently
3. Execute



New EPaxos Benchmarking Strategy

- Reproduced original EPaxos results
  - EPaxos has “optimal” median commit latency
    - 1 WAN RTT to simple majority for 3 and 5 node clusters
- Revised experimental methodology
  - Original did not capture the full impact of conflicts
    - Ex: we measured execution latency, rather than commit latency
      - Results of operations matter
      - Execution latency >= commit latency

Capturing the Impact of Conflicts

	Original Evaluation	Our Evaluation
Primary latency metric	Commit (median)	Execution (mean)
Interfering keys	“Hot key” for all interference	Zipfian distribution
Operation type	All writes	Writes + reads
Access Patterns	3 “hot key” fractions	Grid with 99 configurations
Load for latency measurements	~4-5% max throughput	~80% max throughput
Operation scheduler	Back-to-back	Poisson + cap
Throughput configuration	LAN	WAN

Capturing the Impact of Conflicts

	Original Evaluation	Our Evaluation
Primary latency metric	Commit (median)	Execution (mean)
Interfering keys	"Hot key" for all interference	Zipfian distribution
Operation type	All writes	Writes + reads
Access Patterns	3 "hot key" fractions	Grid with 99 configurations
Load for latency measurements	~4-5% max throughput	~80% max throughput
Operation scheduler	Back-to-back	Poisson + cap
Throughput configuration	LAN	WAN

Capturing the Impact of Conflicts

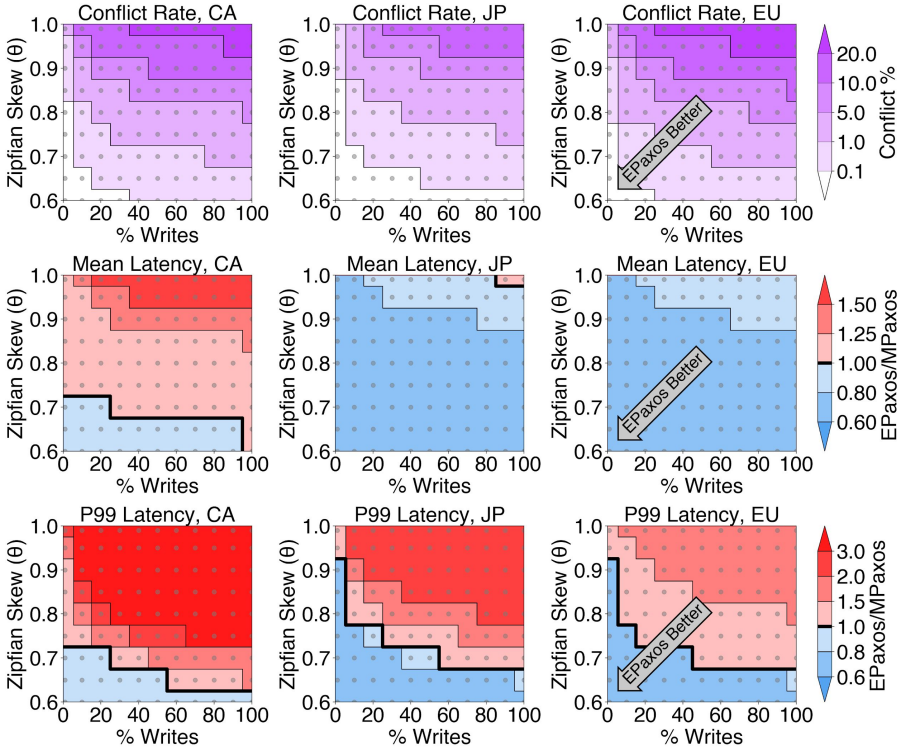
	Original Evaluation	Our Evaluation
Primary latency metric	Commit (median)	Execution (mean)
Interfering keys	“Hot key” for all interference	Zipfian distribution
Operation type	All writes	Writes + reads
Access Patterns	3 “hot key” fractions	Grid with 99 configurations
Load for latency measurements	~4-5% max throughput	~80% max throughput
Operation scheduler	Back-to-back	Poisson + cap
Throughput configuration	LAN	WAN

Capturing the Impact of Conflicts

	Original Evaluation	Our Evaluation
Primary latency metric	Commit (median)	Execution (mean)
Interfering keys	“Hot key” for all interference	Zipfian distribution
Operation type	All writes	Writes + reads
Access Patterns	3 “hot key” fractions	Grid with 99 configurations
Load for latency measurements	~4-5% max throughput	~80% max throughput
Operation scheduler	Back-to-back	Poisson + cap
Throughput configuration	LAN	WAN

EPaxos Evaluation

- Performance varies greatly by workload
  - 0-35% conflict rates
- EPaxos usually improves mean
- EPaxos tail latency worse
  - P99 can be >4x MPaxos

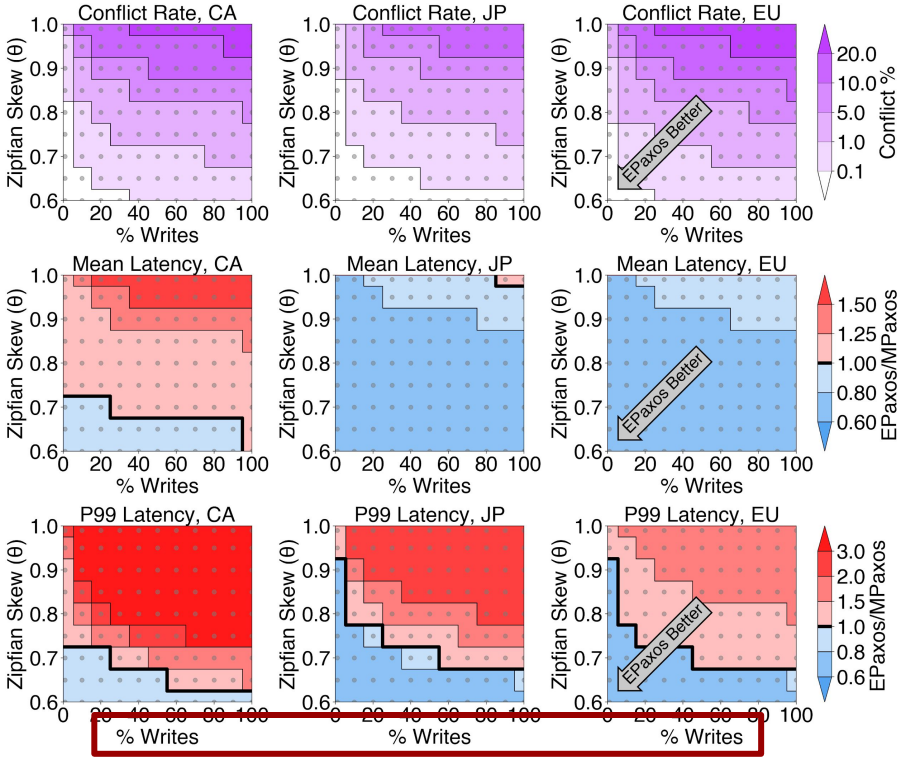


Red = MPaxos better, Blue = EPaxos better

*Nodes OR and VA omitted

EPaxos Evaluation

- Performance varies greatly by workload
  - 0-35% conflict rates
- EPaxos usually improves mean
- EPaxos tail latency worse
  - P99 can be >4x MPaxos

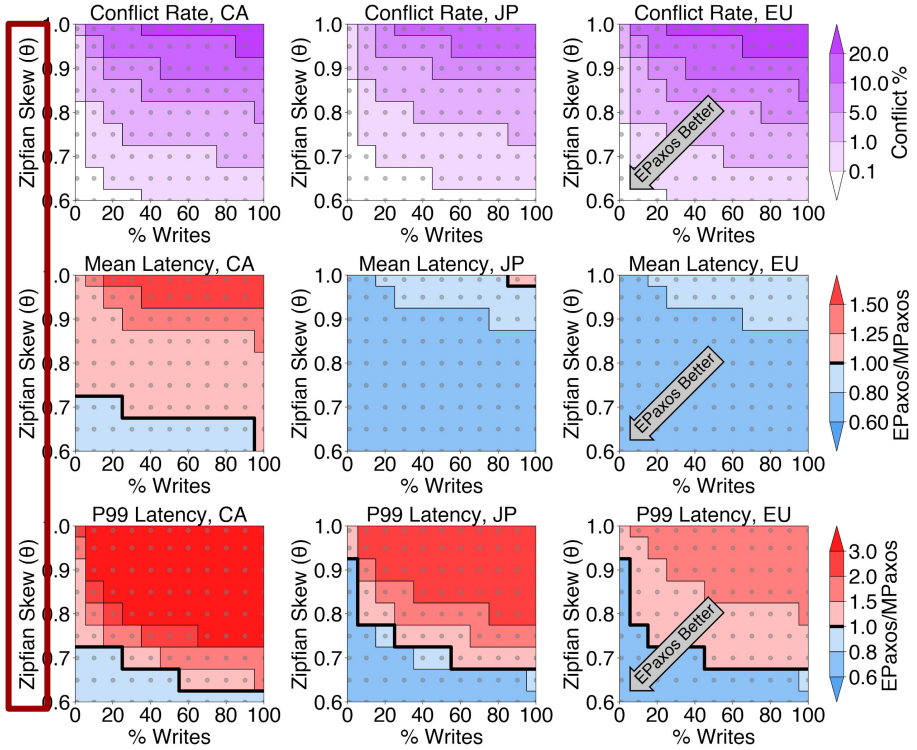


Red = MPaxos better, Blue = EPaxos better

*Nodes OR and VA omitted

EPaxos Evaluation

- Performance varies greatly by workload
  - 0-35% conflict rates
- EPaxos usually improves mean
- EPaxos tail latency worse
  - P99 can be >4x MPaxos

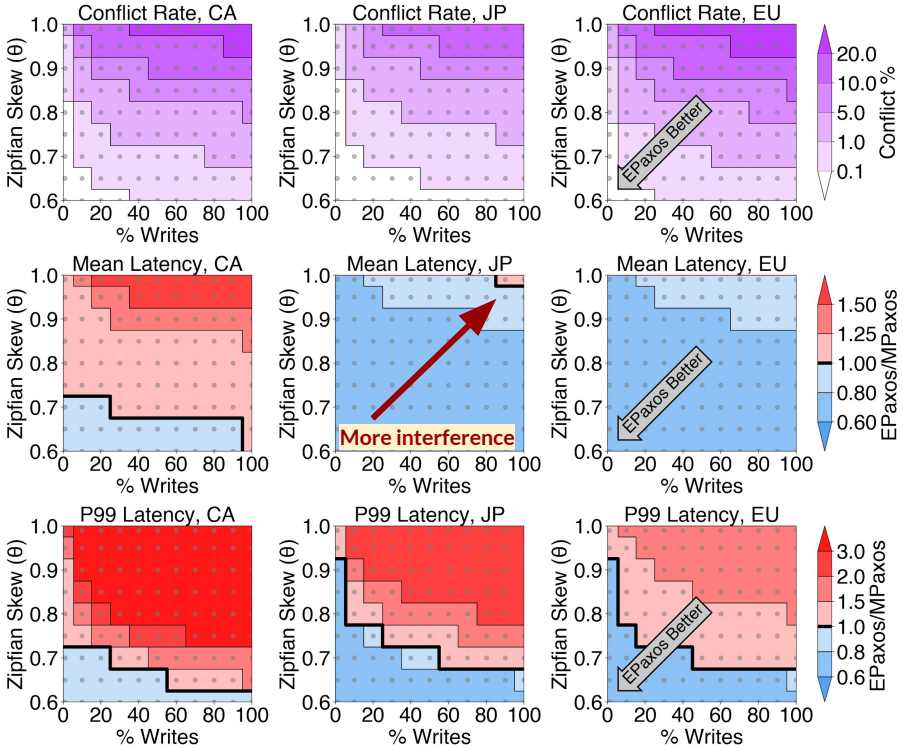


Red = MPaxos better, Blue = EPaxos better

*Nodes OR and VA omitted

EPaxos Evaluation

- Performance varies greatly by workload
  - 0-35% conflict rates
- EPaxos usually improves mean
- EPaxos tail latency worse
  - P99 can be >4x MPaxos

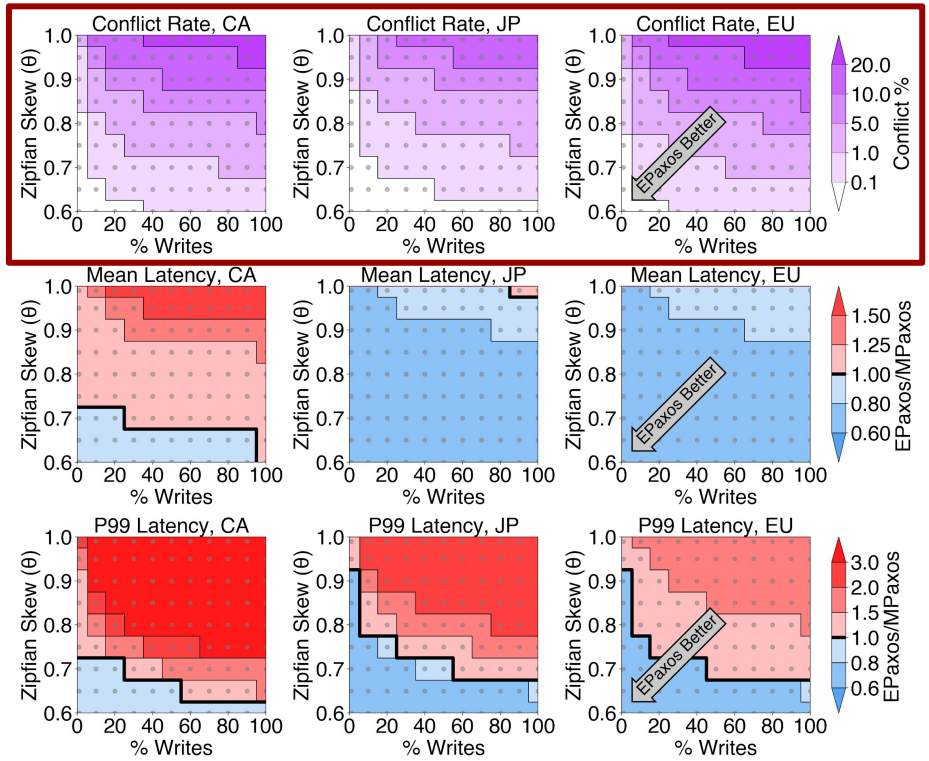


Red = MPaxos better, Blue = EPaxos better

*Nodes OR and VA omitted

EPaxos Evaluation

- Performance varies greatly by workload
  - 0-35% conflict rates
- EPaxos usually improves mean
- EPaxos tail latency worse
  - P99 can be >4x MPaxos

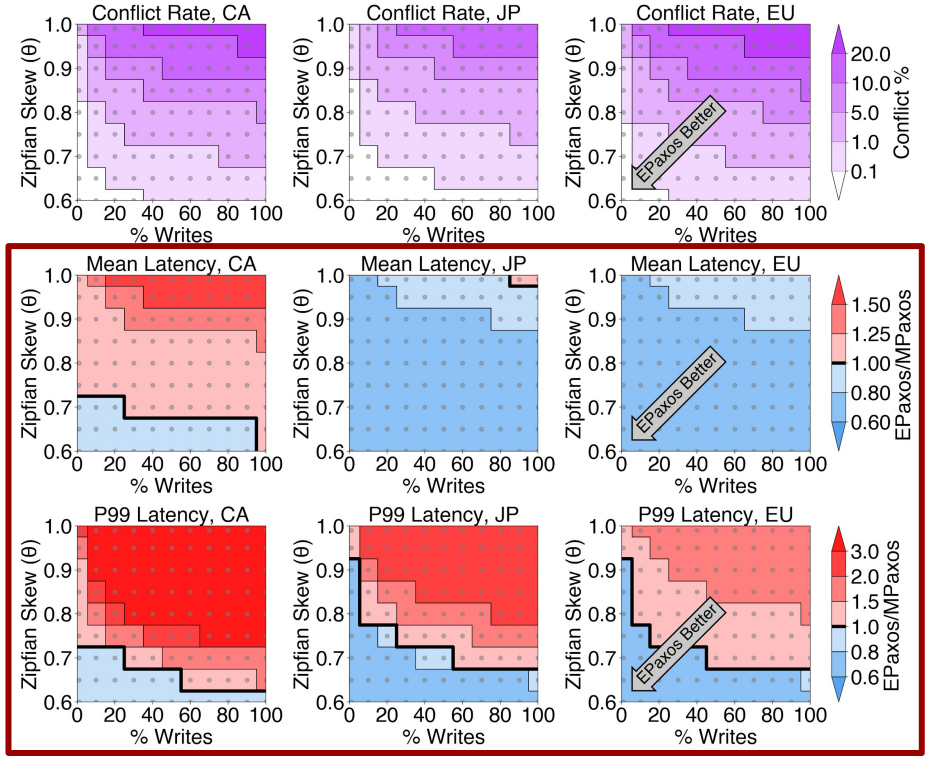


Red = MPaxos better, Blue = EPaxos better

*Nodes OR and VA omitted

EPaxos Evaluation

- Performance varies greatly by workload
  - 0-35% conflict rates
- EPaxos usually improves mean
- EPaxos tail latency worse
  - P99 can be >4x MPaxos

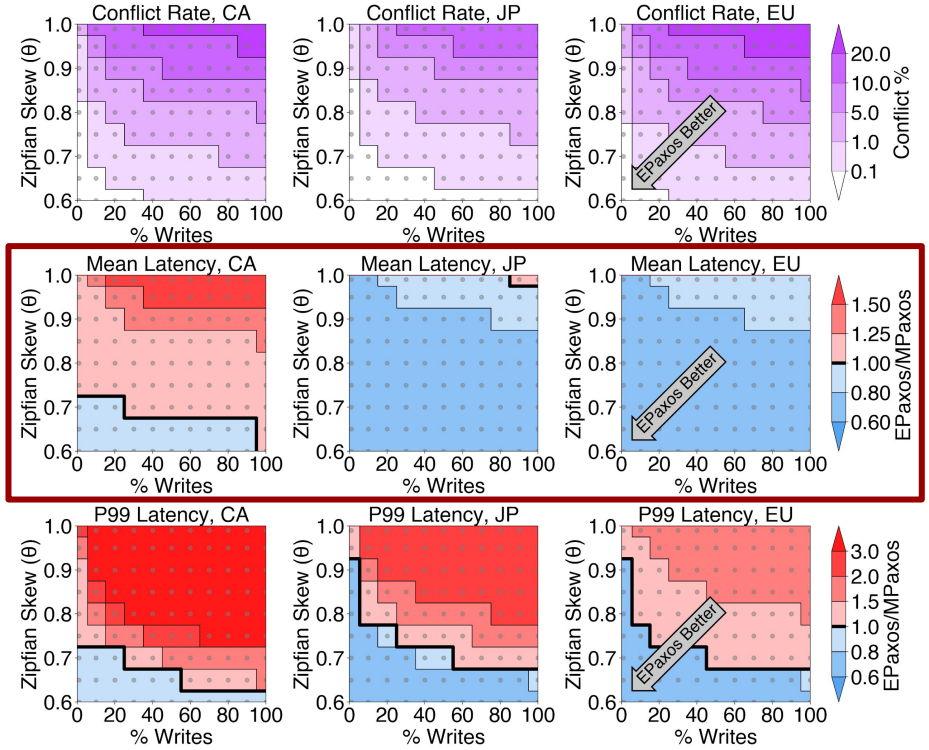


Red = MPaxos better, Blue = EPaxos better

*Nodes OR and VA omitted

EPaxos Evaluation

- Performance varies greatly by workload
  - 0-35% conflict rates
- EPaxos usually improves mean
- EPaxos tail latency worse
  - P99 can be >4x MPaxos

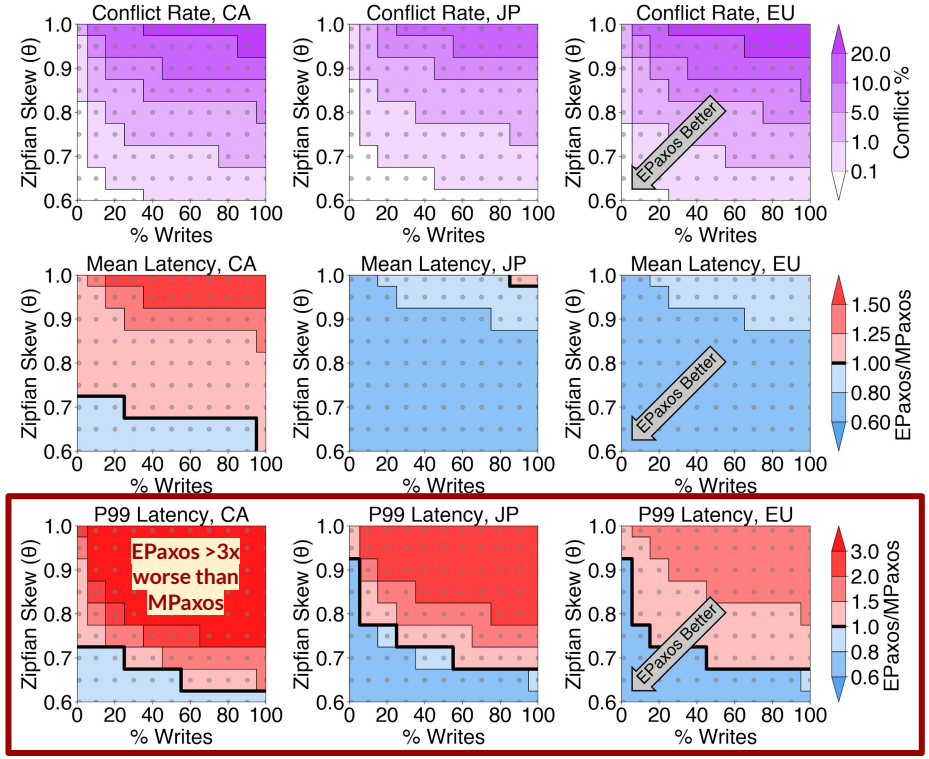


Red = MPaxos better, Blue = EPaxos better

*Nodes OR and VA omitted

EPaxos Evaluation

- Performance varies greatly by workload
  - 0-35% conflict rates
- EPaxos usually improves mean
- EPaxos tail latency worse
  - P99 can be >4x MPaxos



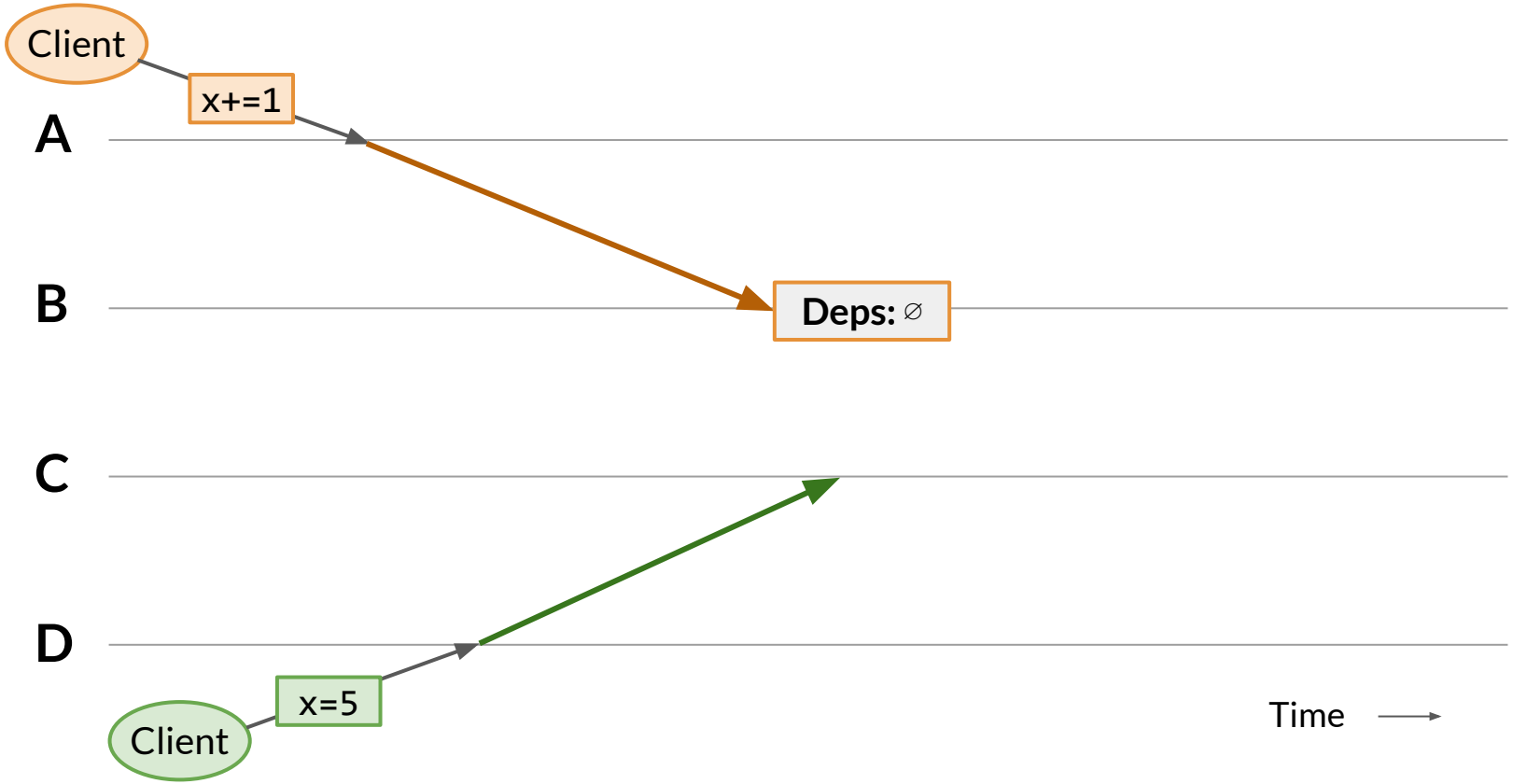
Red = MPaxos better, Blue = EPaxos better

*Nodes OR and VA omitted

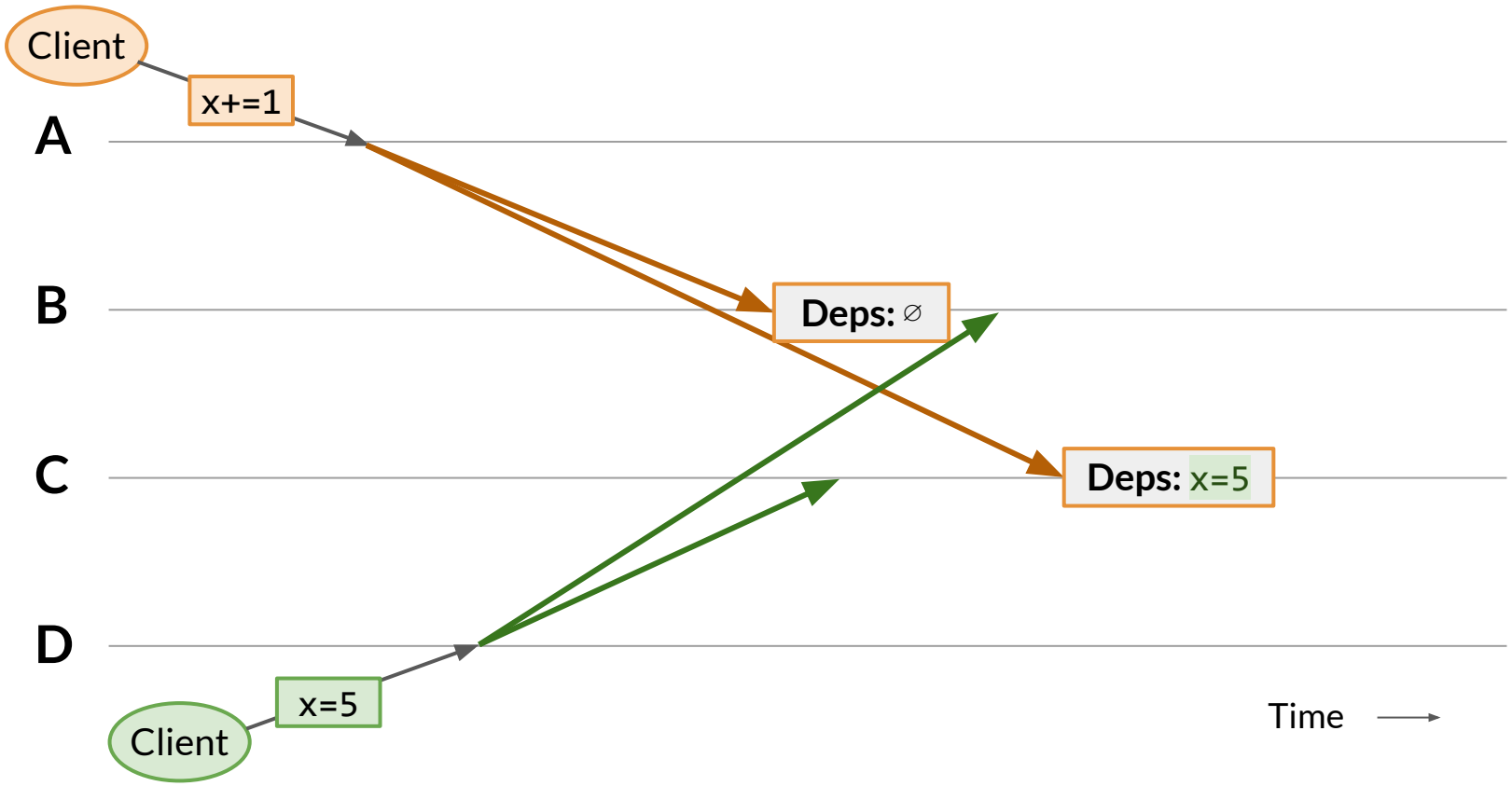
Reducing Conflicts

- Conflicts increase latency
- Can we reduce conflicts?
  - Timestamp-Ordered Queuing (TOQ)
  - Delay message processing
  - Use synchronized clocks

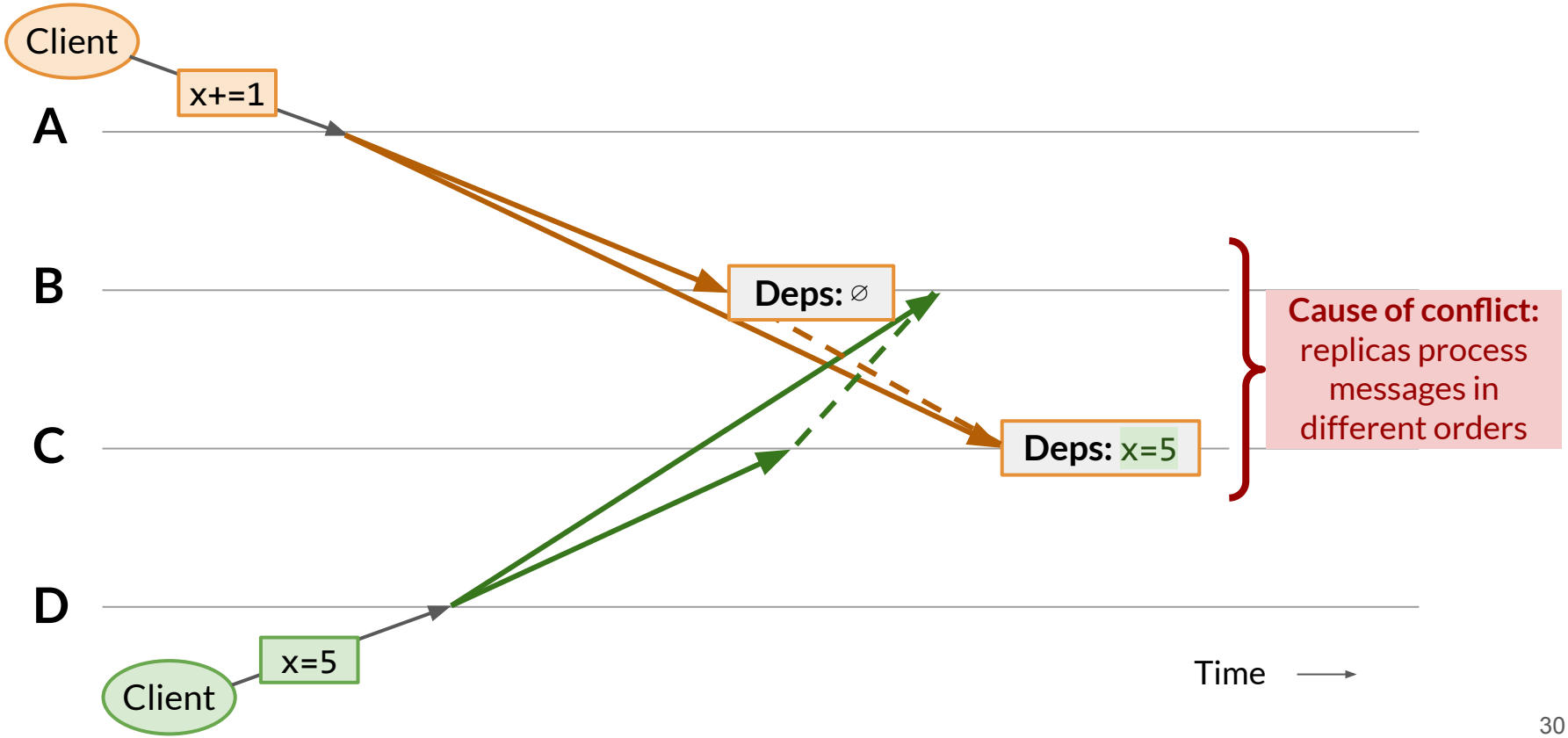
TOQ Intuition



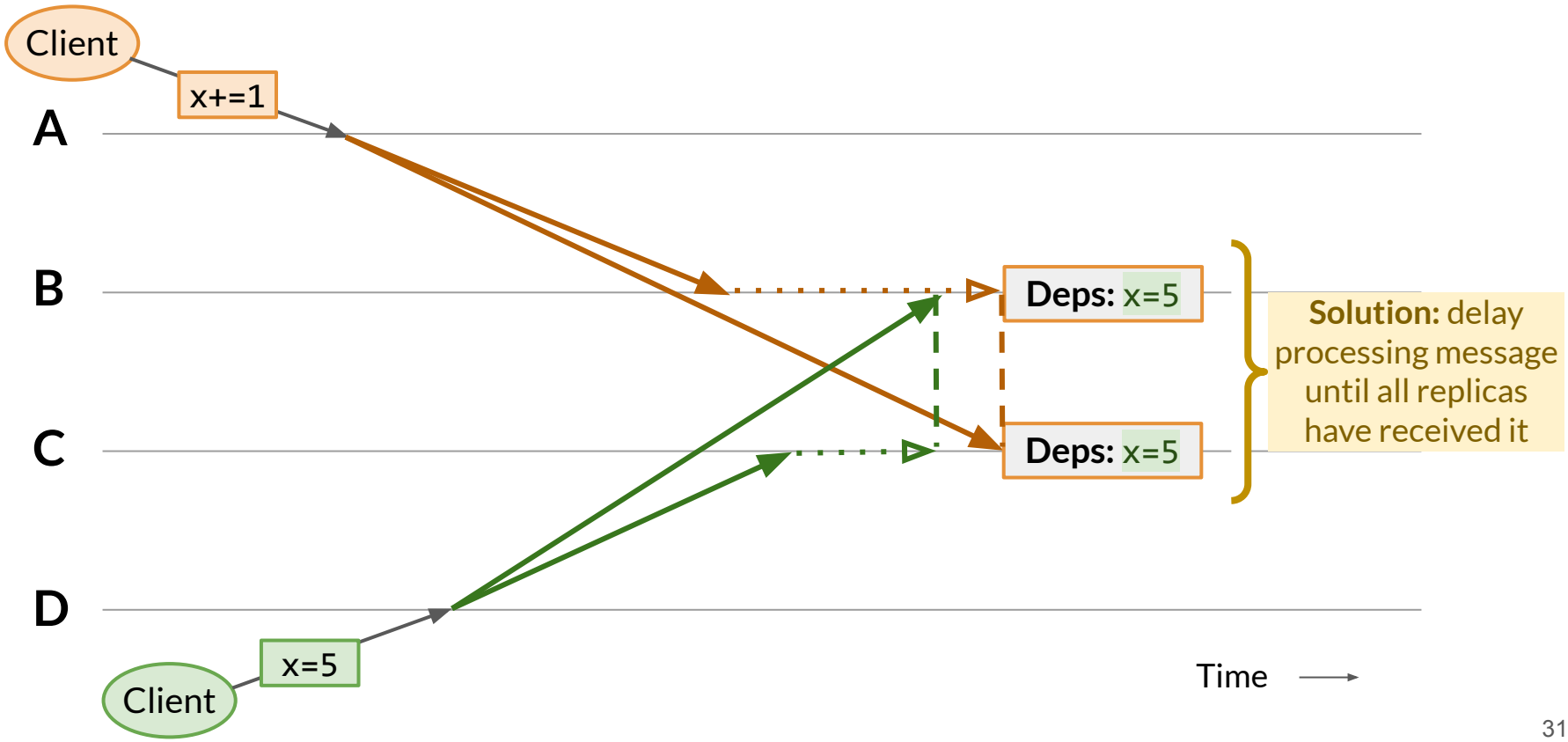
TOQ Intuition



TOQ Intuition

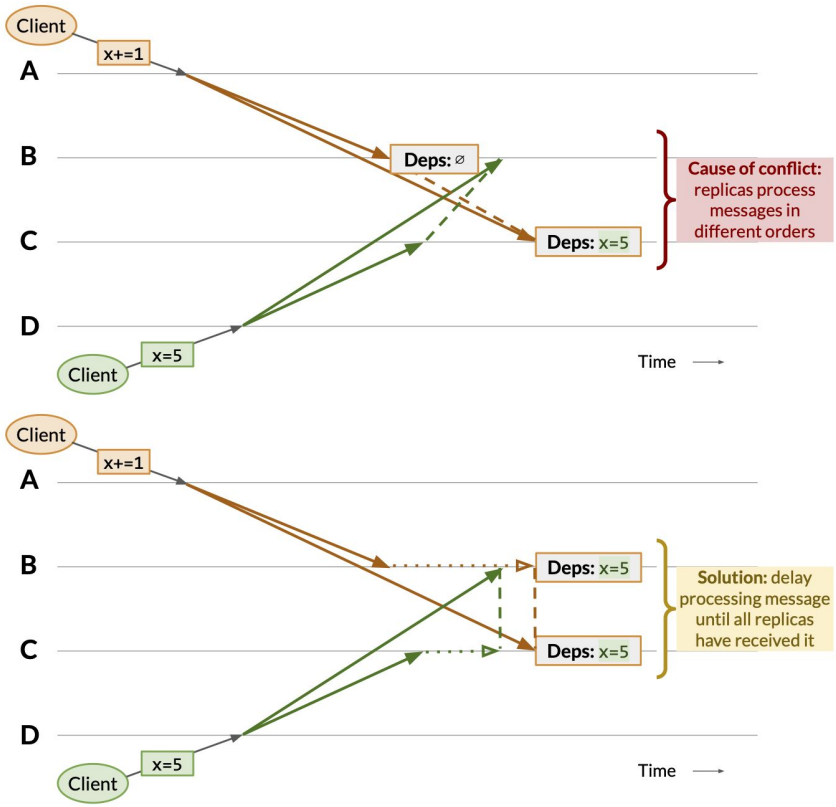


TOQ Intuition



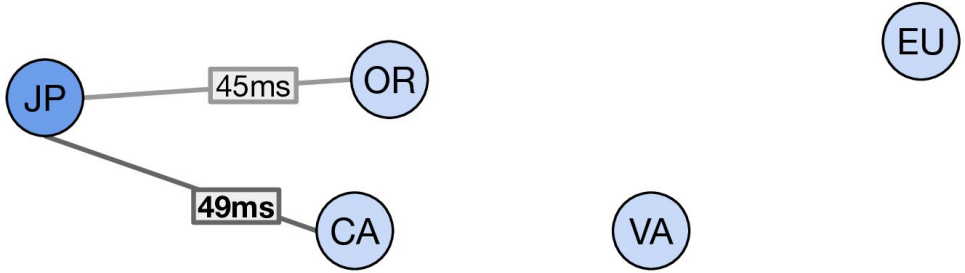
Timestamp-Ordered Queueing (TOQ)

- **Goal:** replicas process new requests at the same time
- How?
  1. Assign a *ProcessAt* time to each request
     - The time the furthest replica should receive it
     - Measure the one-way delays between replicas
  2. Used synchronized clocks to implement

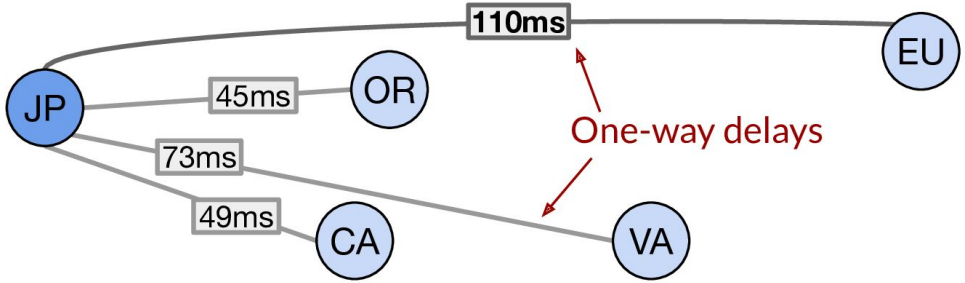


Fully Consistent TOQ Increases Latency

- Without TOQ
  - ***Quorum*** propagation delay bottlenecks latency

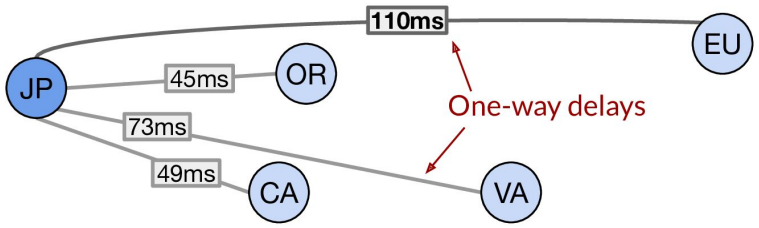


- With TOQ
  - ***Cluster*** propagation delay bottlenecks latency
  - +61ms for JP in our test cluster

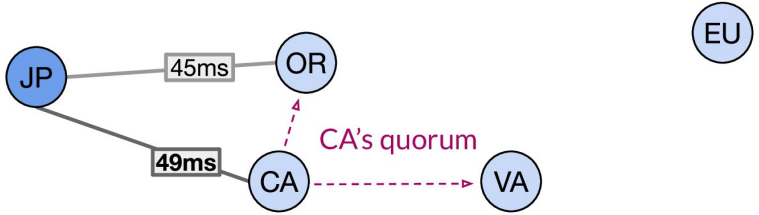


TOQ Sync Groups: Conflict vs. Latency Trade-off

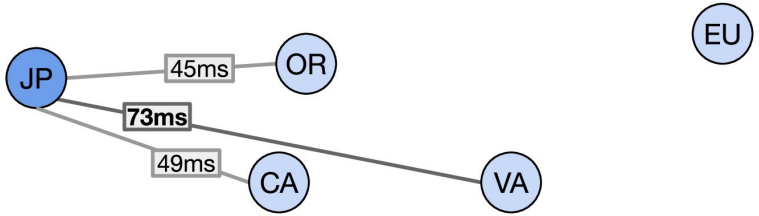
- **Synchronize all**
  - *Problem*: no one can respond until furthest replica receives message



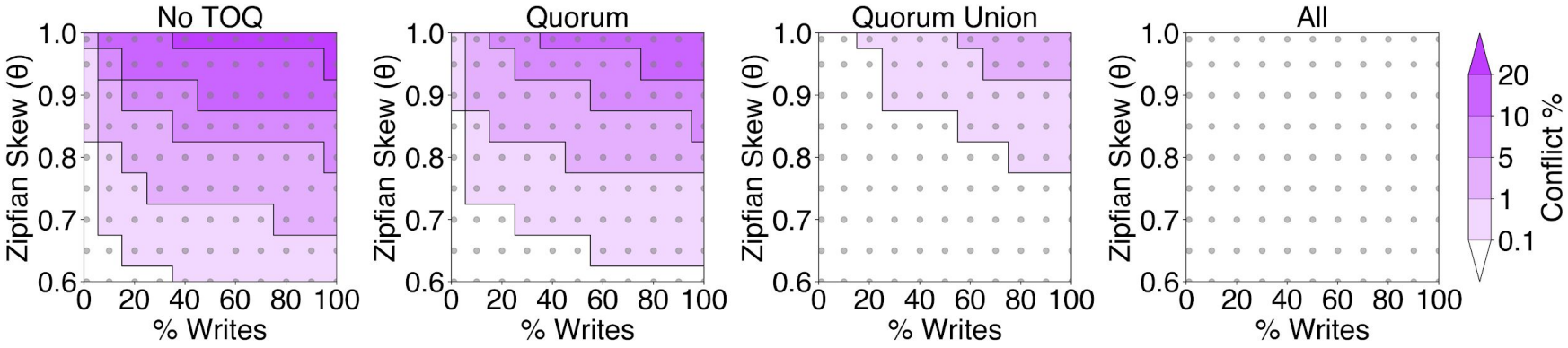
- **Synchronize quorum**
  - *Problem*: some replicas out of sync
    - Conflicts because different replicas have different quorums



- **Hybrid: quorum union**
  - Some increase in best-case latency for some replicas



TOQ Results



Quorum

- **50%+** reduction in conflicts
- Up to **7.5%** decrease in mean latency

Quorum Union

- **85%+** reduction in conflicts
- Up to -14% mean latency
- One location +24% best-case latency

All

- **100%** reduction in conflicts
- One location +62% best-case latency

*Graphs for replica in OR

Additional Content

- EPaxos livelock
  - Execution latency unbounded in original protocol
  - We bounded it at ~3 WAN RTTs
- Batching impact on latency
- Throughput analysis

Conclusion

- Conflicts hinder EPaxos performance
  - Developed more comprehensive experimental methodology
  - Confirmed that EPaxos reduces median latency
    - 30% lower than MPaxos
  - Slow path worse than originally measured
    - >4x increase in 99th percentile latency
- Can use synchronized clocks to reduce conflict rate
  - TOQ effective without increasing best-case latency
    - Reduces EPaxos conflict rate by at least 50%
    - Reduces EPaxos mean latency by up to 7.5%

Thank you!

Questions? stollman@cs.stanford.edu