



UCDAVIS

RESILIENTDB: BUILDING GLOBAL-SCALE PRIVACY-PRESERVING BLOCKCHAIN FABRIC

Mohammad Sadoghi

REIMAGINE 2020
Global Blockchain Conference
May 18, 2020



Mohammad Sadoghi
Exploratory Systems Lab
Department of Computer Science
UCDAVIS
UNIVERSITY OF CALIFORNIA





ExpoLab Team



Mohammad Sadoghi
(Principal Investigator)



Jelle Hellings, PostDoc
(Fault-tolerant Complexity Analysis)



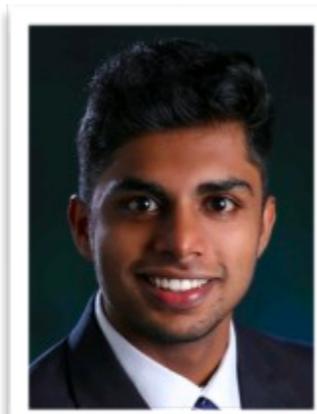
Suyash Gupta, PhD
(Scalable Consensus Meta-Protocols)



Thamir Qadah, PhD
(Distributed & Coordination-free Concurrency)



Sajjad Rahnama, PhD
(Global Scale Consensus)



Dhruv Krishnan, MSc
(Scaling Fabric via Sharding)



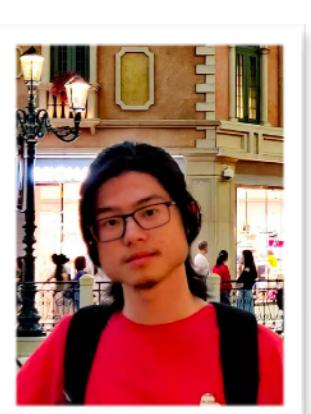
Priya Holani, MSc
(Scaling Fabric via Sharding)



Shubham Pandey, MSc
(Scaling Fabric via RDMA)



Rohan Sogani, MSc
(Scaling Fabric via Sharding)

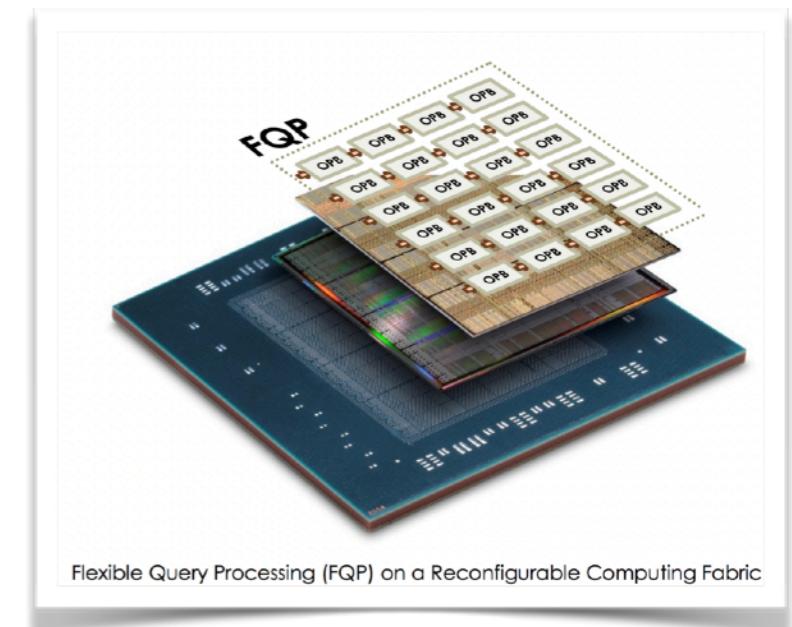


Xinyuan Sun, MSc
(Scaling Fabric via RDMA)

Journey...



**SQL
Analytics**

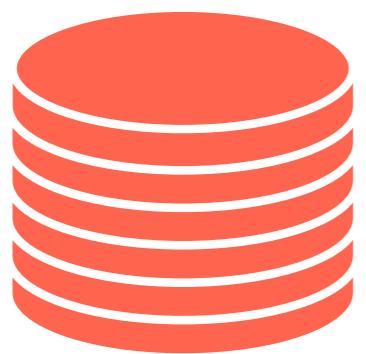


Flexible Query Processing (FQP) on a Reconfigurable Computing Fabric

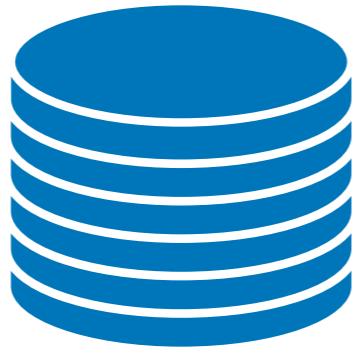
FPGA Acceleration: FQP (Flexible Query Processor)

[VLDB'10, ICDE'12, VLDB'13, ICDE'15, SIGMOD Record'15, ICDE'16, USENIX ATC'16, ICDCS'17, ICDE'18, TKDE'19]

Journey...



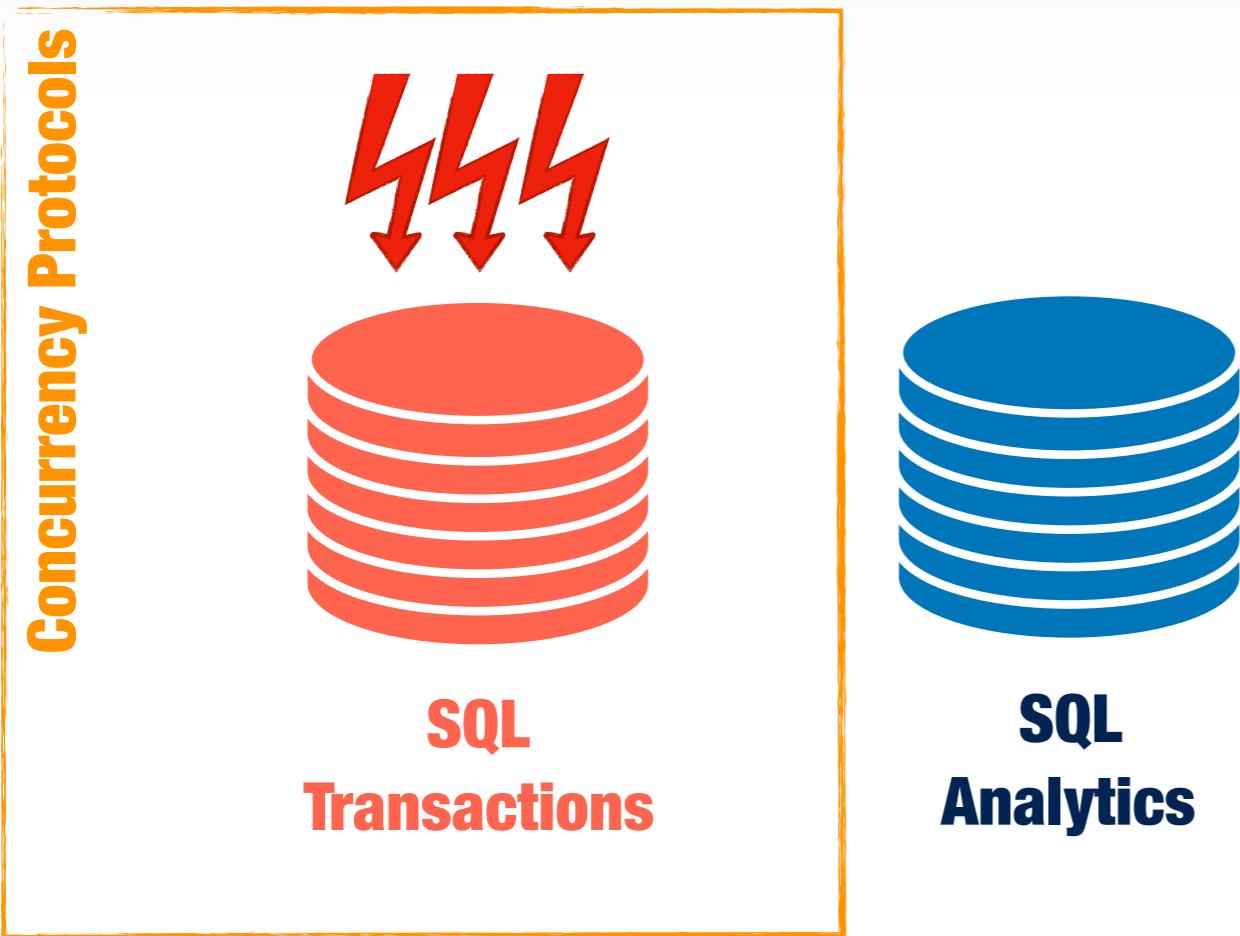
**SQL
Transactions**



**SQL
Analytics**

High-dimensional Indexing: (e.g., BE-Tree, BE-topK)
[SIGMOD'11, ICDE'12, TODS'13, ICDCS'13, ICDE'14, ICDCS'17, Middleware'17]

Journey...

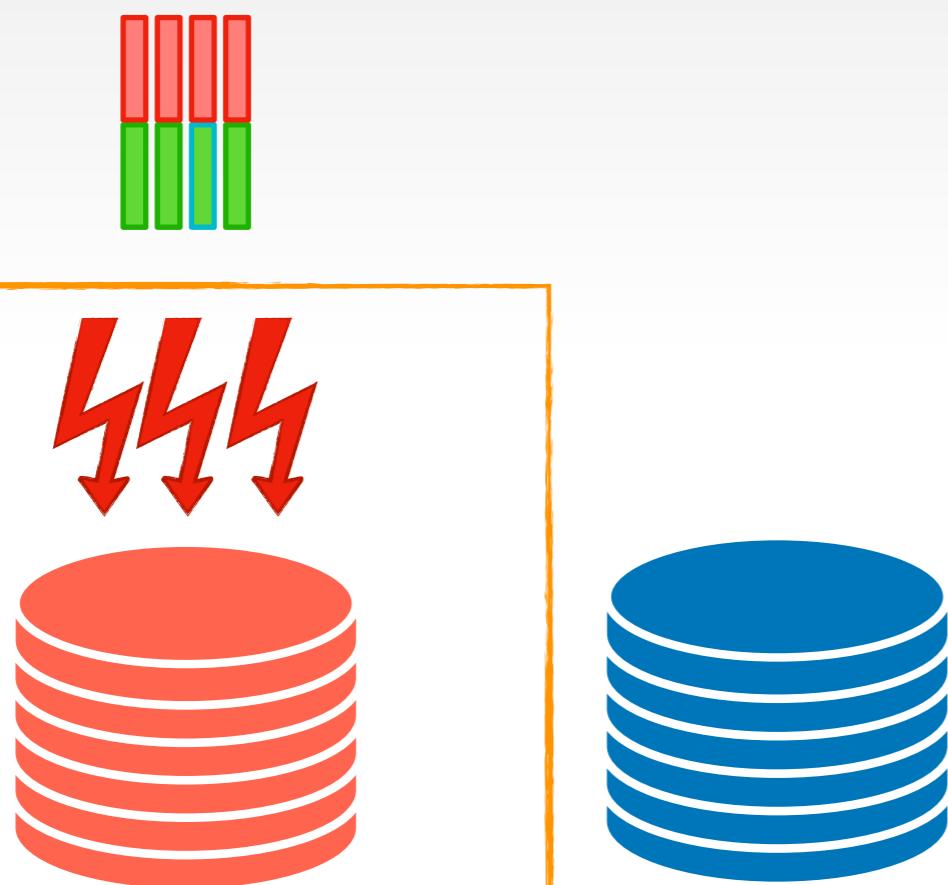


Concurrency Control Protocols: (e.g., 2VCC, QueCC - Best Paper Award)
[VLDB'13, VLDB'14, VLDBJ'16, Middleware'16, TDKE'15, SIGMOD'15, ICDE'16, Middleware'18]

Journey...

QueCC: Queue-Oriented Planning and Execution Architecture

Concurrency Protocols

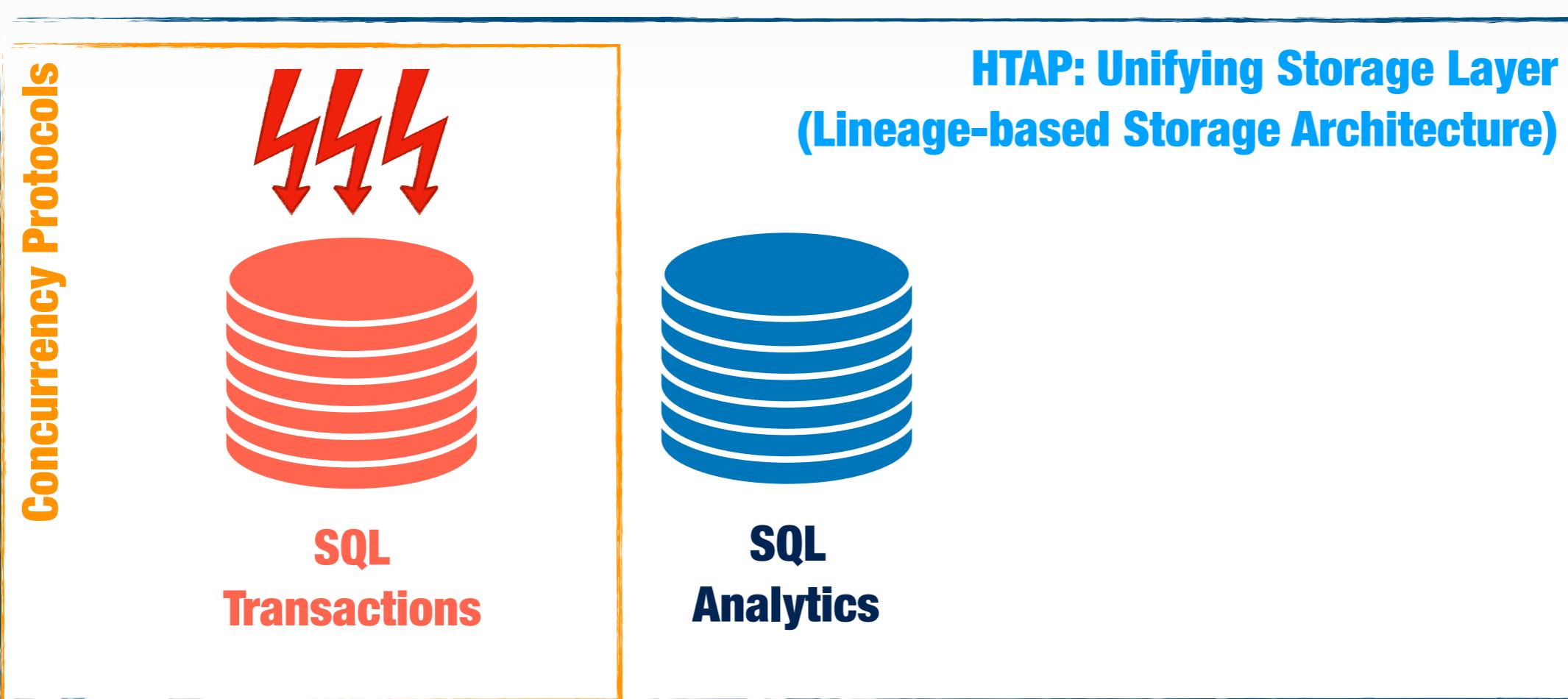


SQL
Transactions

SQL
Analytics

Concurrency Control Protocols: (e.g., 2VCC, QueCC - Best Paper Award)
[VLDB'13, VLDB'14, VLDBJ'16, Middleware'16, TDKE'15, SIGMOD'15, ICDE'16, Middleware'18]

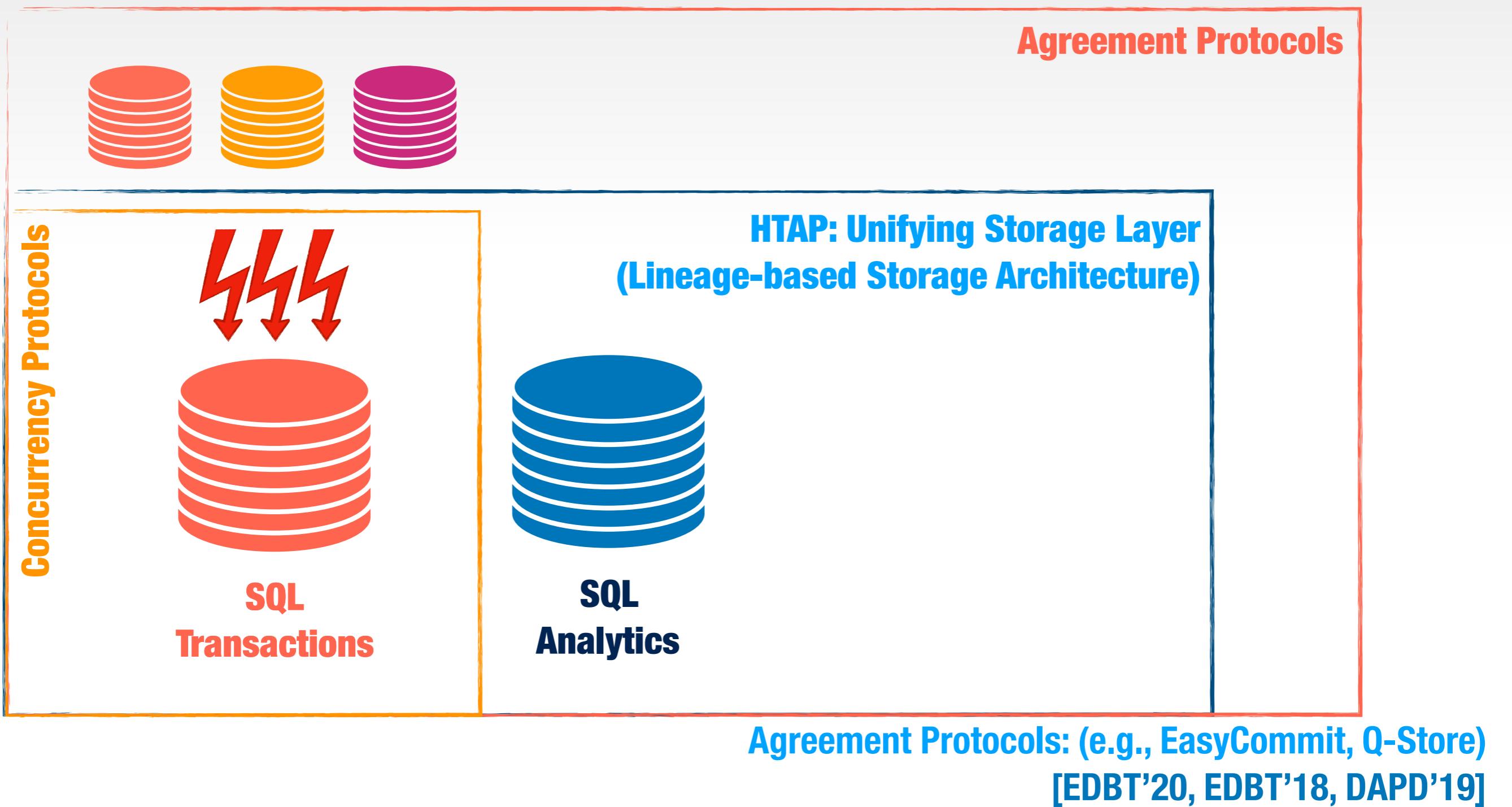
Journey...



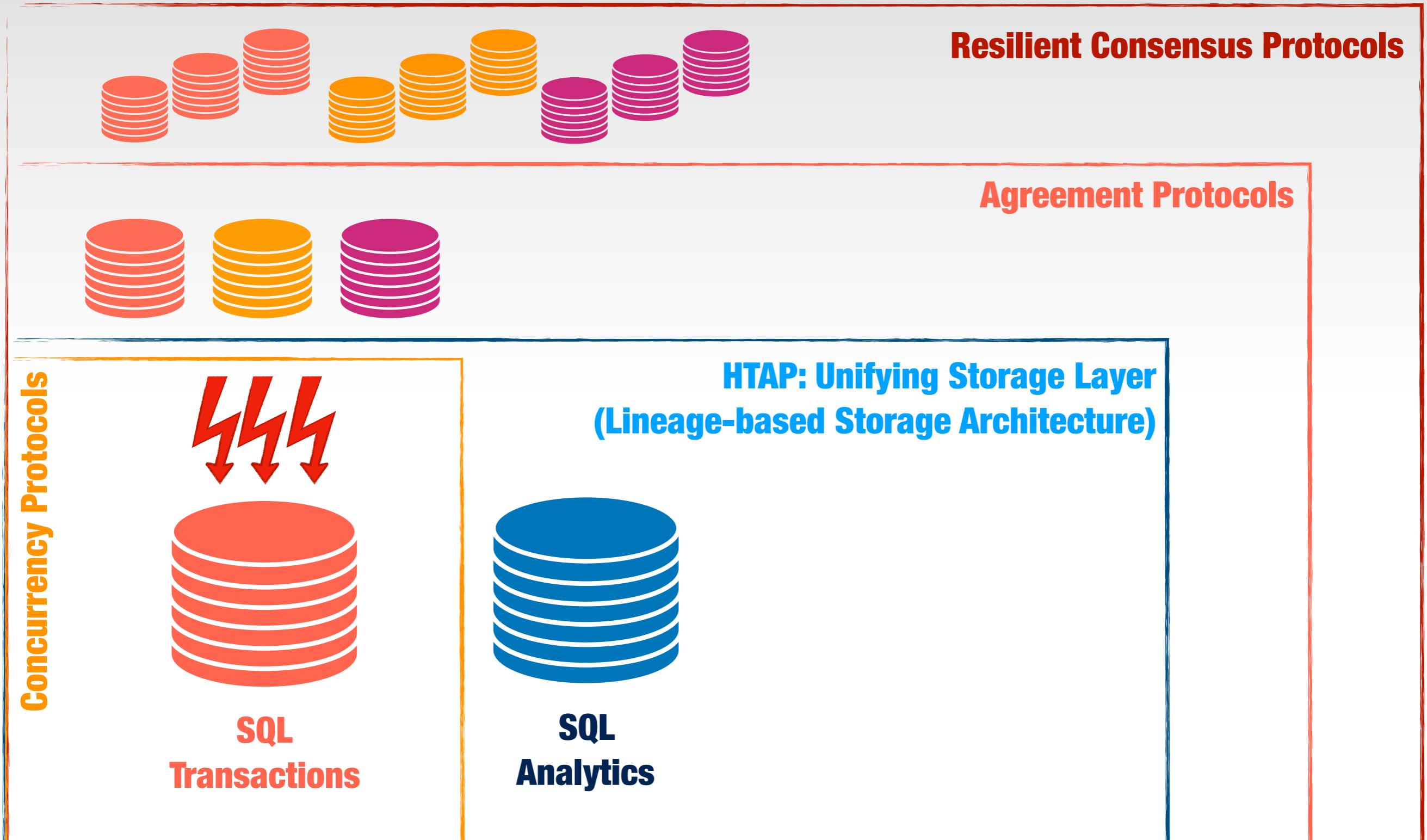
**HTAP Column-store: L-Store (Lineage-based Data Store)
[VLDB'12, ICDE'14, ICDCS'16, EDBT'18, 34 filed US patents]**

Graphs on SQL: (e.g., GRFusion) [SIGMOD'18, EDBT'18] 7

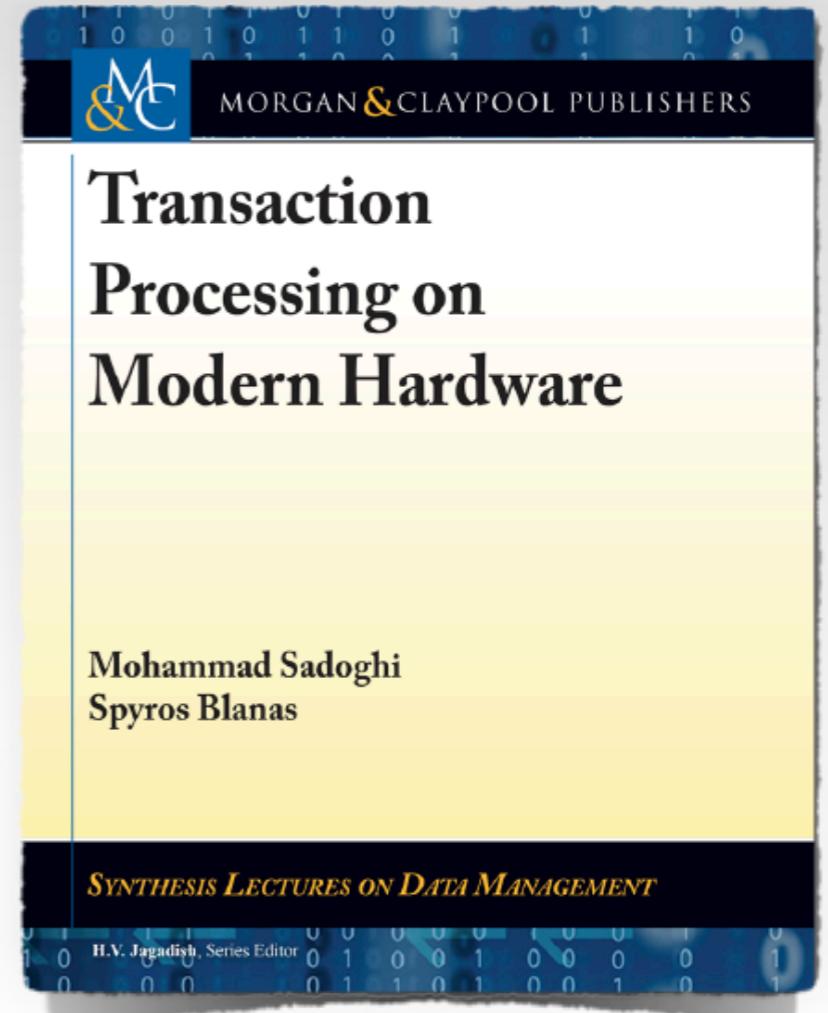
Journey...



Journey...



Consensus Protocols: (e.g., ResilientDB, Blockplane, Blocklite, Delayed Replication, CSP)
[VLDB'20, ICDCS'20, ICDT'20, SC'19, ICDE'19 (2x), DISC'19 (2x), arXiv'19 (6x)]



Books

Transaction Processing on Modern Hardware.

Synthesis Lectures on Data Management, Morgan & Claypool Publishers 2019

Fault-Tolerant Distributed Transactions on Blockchain.

Synthesis Lectures on Data Management, Morgan & Claypool Publishers, *to appear* 2020



Press

Advancements TV With Ted Danson - CNBC, CityAM, Medium, Yahoo! Finance, Market Insider, CoinDesk, Crypto Media, Davis Enterprise, Times Union, WBOC TV/Radio

Books

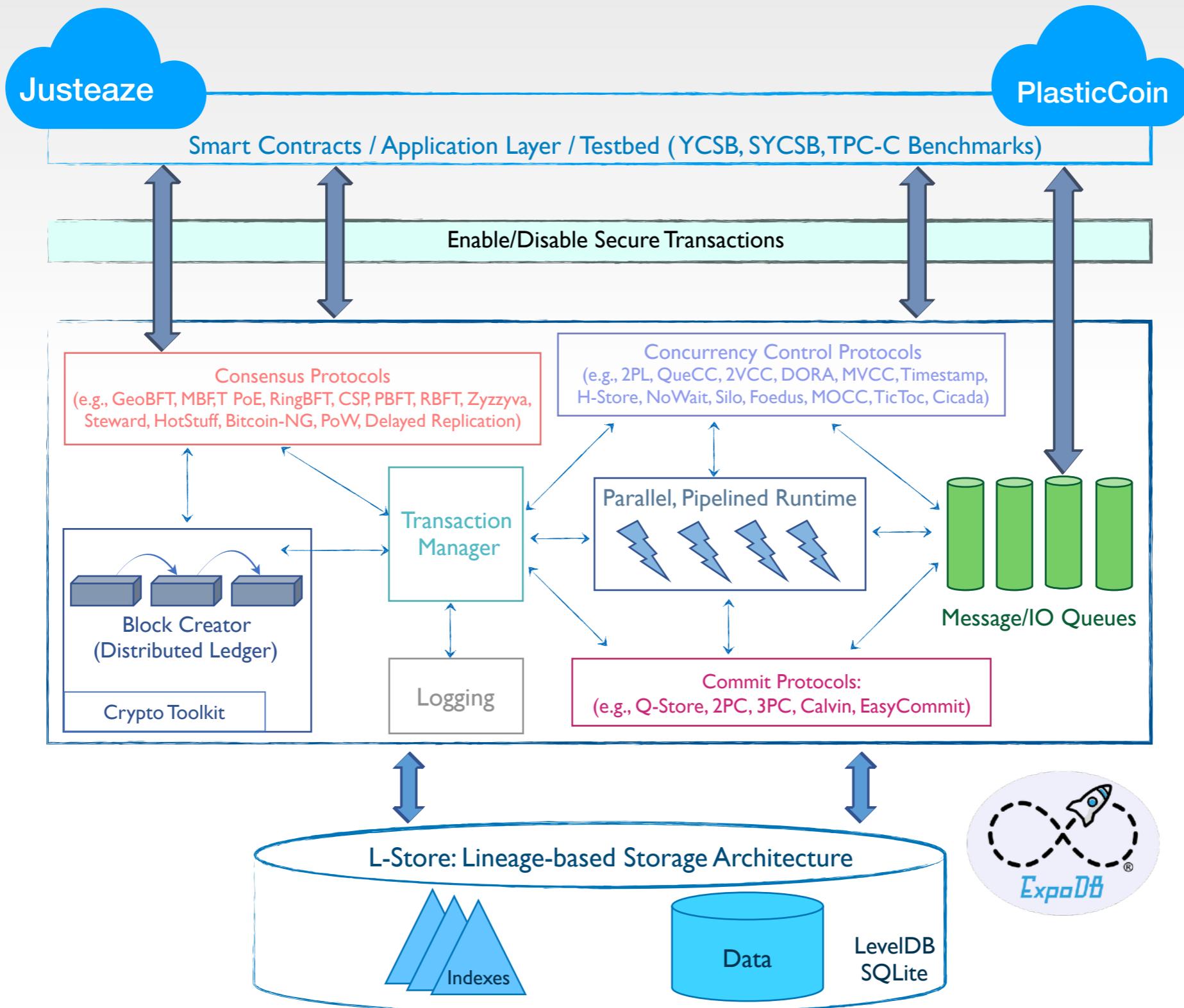
Transaction Processing on Modern Hardware.

Synthesis Lectures on Data Management, Morgan & Claypool Publishers 2019

Fault-Tolerant Distributed Transactions on Blockchain.

Synthesis Lectures on Data Management, Morgan & Claypool Publishers, *to appear* 2020

ExpoDB Architecture





Quantifiable Resiliency (Graduate Student Experiments)

Aloha Lake, Desolation Wilderness
15 Miles Long
2,500 Feet Elevation Gain
(8,700 Feet at Summit)



Tomales Point Trail, Point Reyes National Seashore

**9.4 Miles Long
1,579 Feet Elevation Gain**



Non-Quantifiable Resiliency

Proof-of-Execution: Reaching Consensus Through Fault-Tolerant Speculation [arXiv'19]

Out-of-Order message processing to reduce replica idleness

Speculative Execution with revertible/divergent replicas &

Speculative Execution after a transition

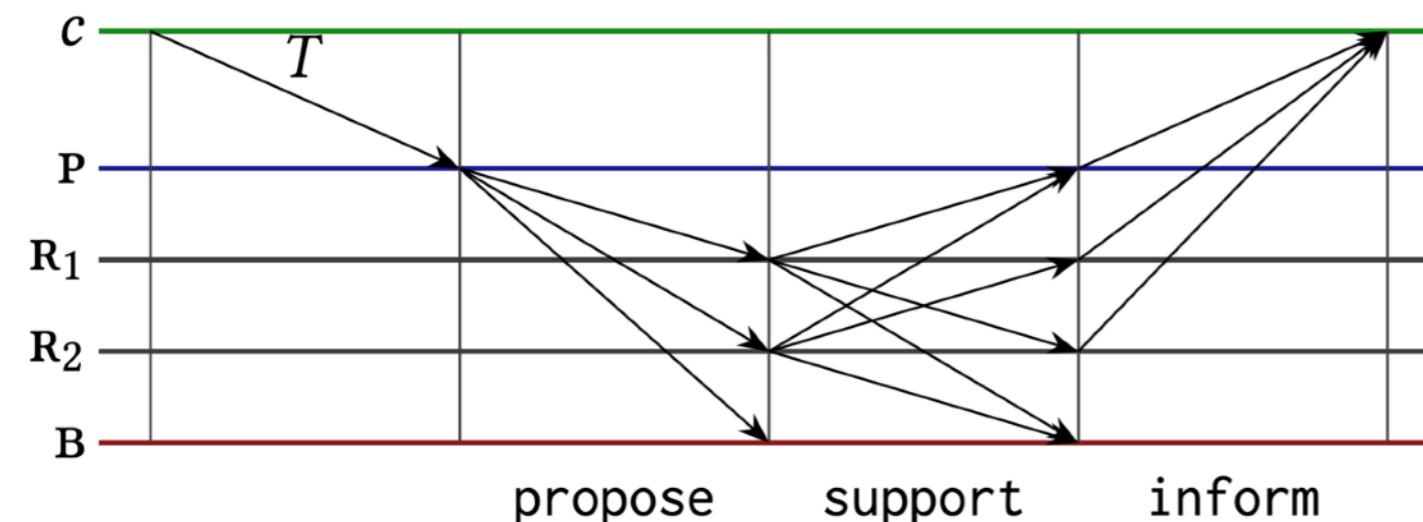
Proof-of-Execution: Reaching Consensus Through Fault-Tolerant Speculation [arXiv'19]

Out-of-Order message processing to reduce replica idleness

Speculative Execution with revertible/divergent replicas & eager/irrevertible client commit

Proof-of-Execution: Reaching Consensus Through Fault-Tolerant Speculation [arXiv'19]

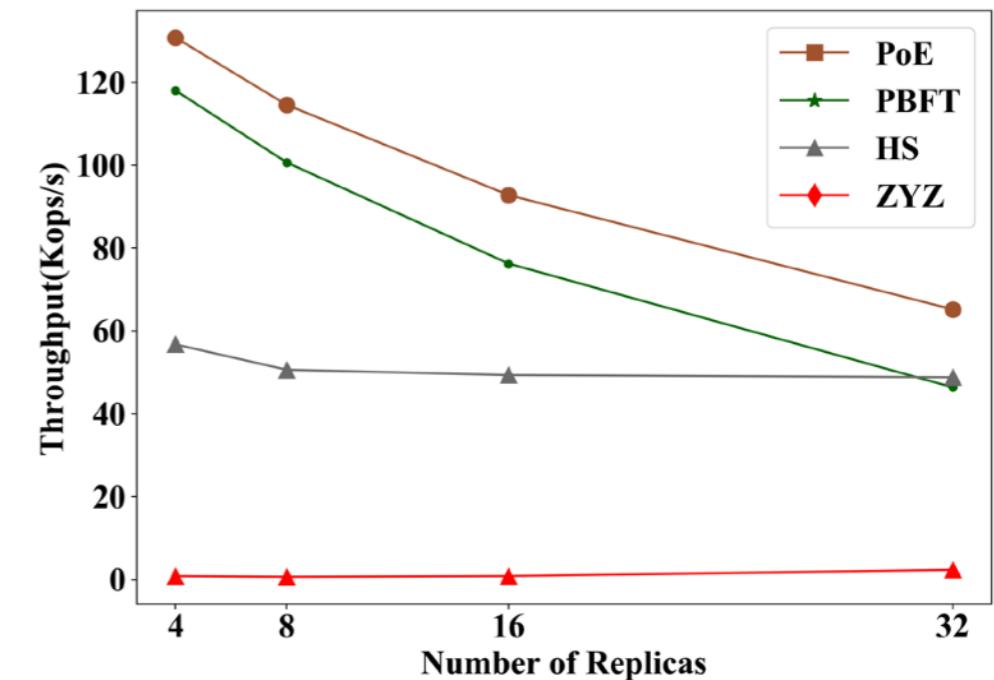
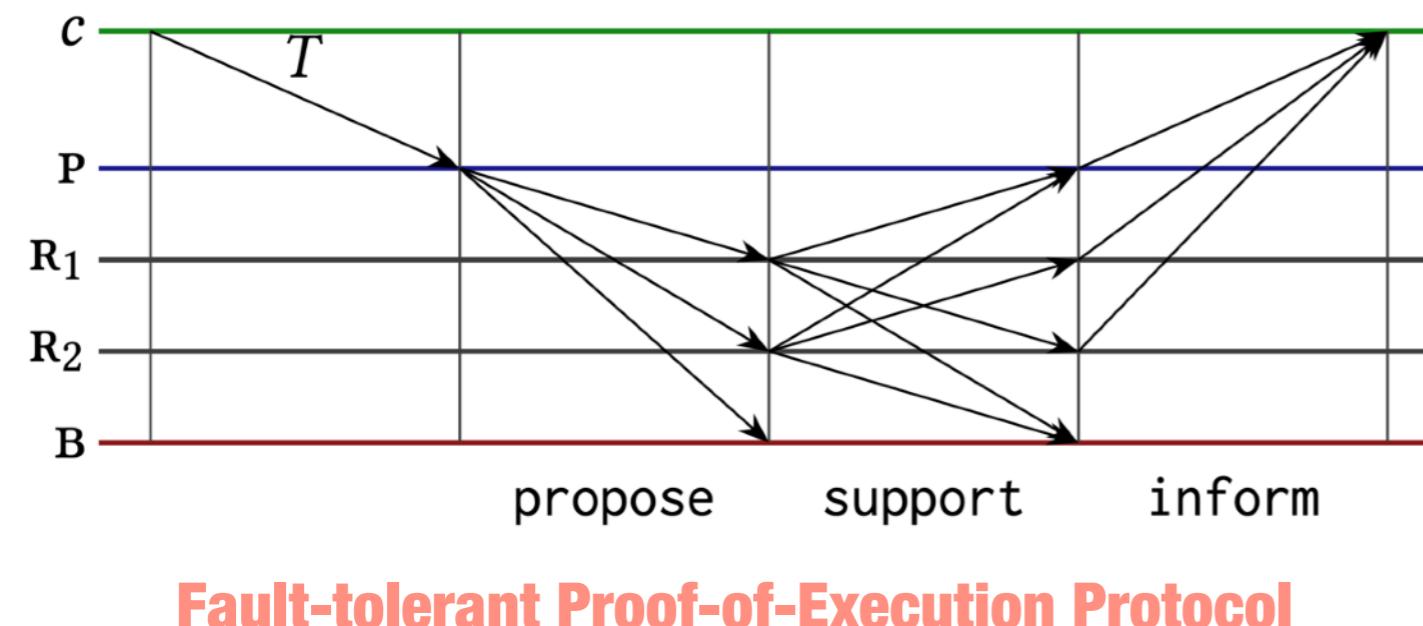
Out-of-Order message processing to reduce replica idleness
Speculative Execution with revertible/divergent replicas &
eager/irrevertible client commit



Fault-tolerant Proof-of-Execution Protocol

Proof-of-Execution: Reaching Consensus Through Fault-Tolerant Speculation [arXiv'19]

Out-of-Order message processing to reduce replica idleness
Speculative Execution with revertible/divergent replicas &
eager/irrevertible client commit



PoE scales beyond 32 replicas, in presence of failures, outperforms PBFT up to 40%

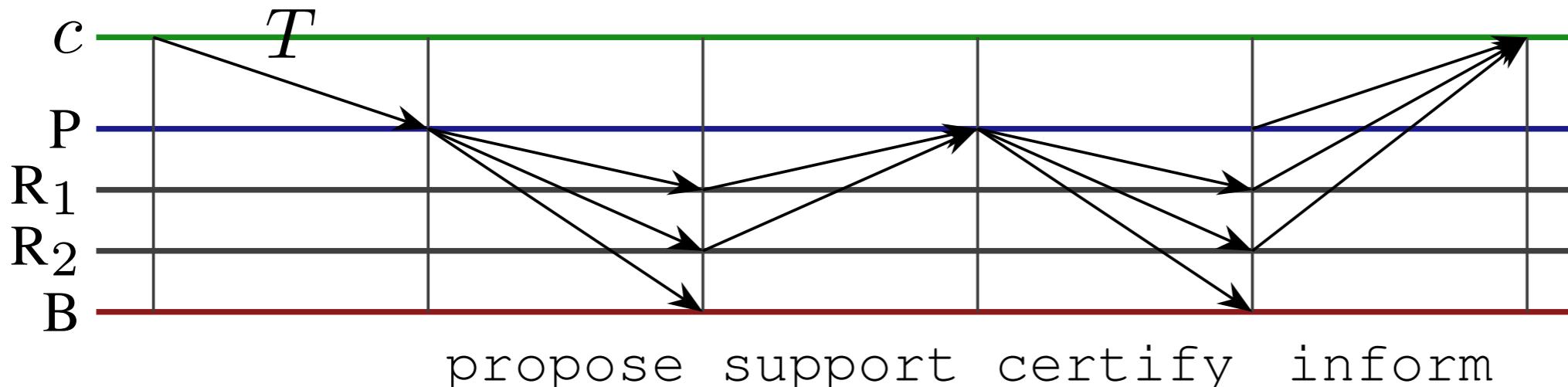
Proof-of-Execution: Reaching Consensus Through Fault-Tolerant Speculation [arXiv'19]

Out-of-Order message processing to reduce replica idleness

Speculative Execution with revertible/divergent replicas &

eager/irrevertible client commit

introducing linear message complexity



Linearized Proof-of-Execution Protocol

MultiBFT: Scaling Blockchain Databases Through Parallel Resilient Consensus Paradigm [arXiv'19]

A wait-free meta-protocol...

Designate multiple replicas as Primaries!

Run multiple parallel consensus on each replica independently

MultiBFT: Scaling Blockchain Databases Through Parallel Resilient Consensus Paradigm [arXiv'19]

A wait-free meta-protocol...

Designate multiple replicas as Primaries!

MultiBFT: Scaling Blockchain Databases Through Parallel Resilient Consensus Paradigm [arXiv'19]

A wait-free meta-protocol...

Designate multiple replicas as Primaries!

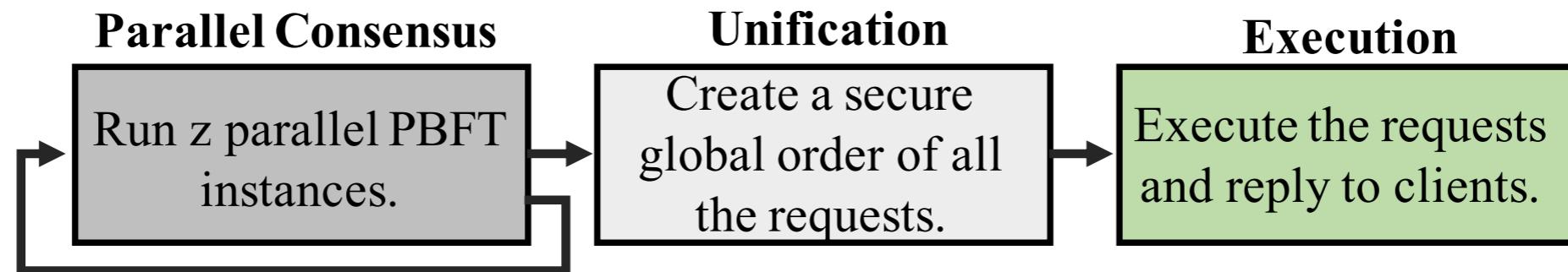
Run multiple parallel consensuses on each replica independently

MultiBFT: Scaling Blockchain Databases Through Parallel Resilient Consensus Paradigm [arXiv'19]

A wait-free meta-protocol...

Designate multiple replicas as Primaries!

Run multiple parallel consensuses on each replica independently

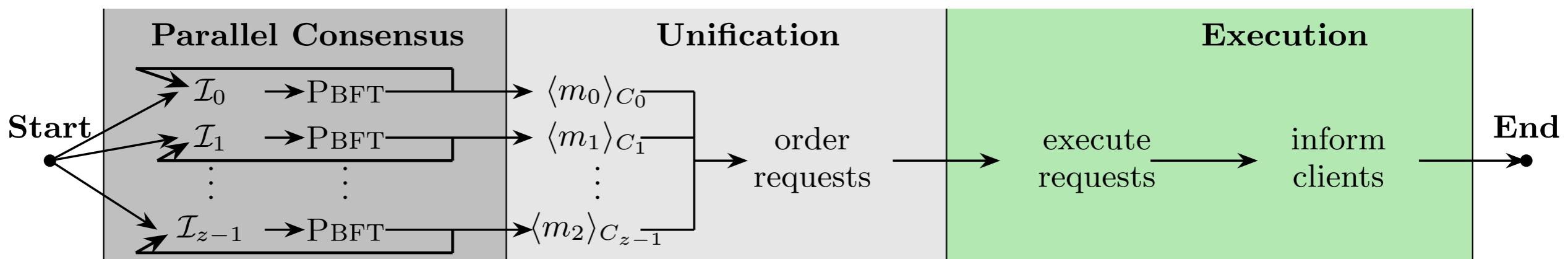
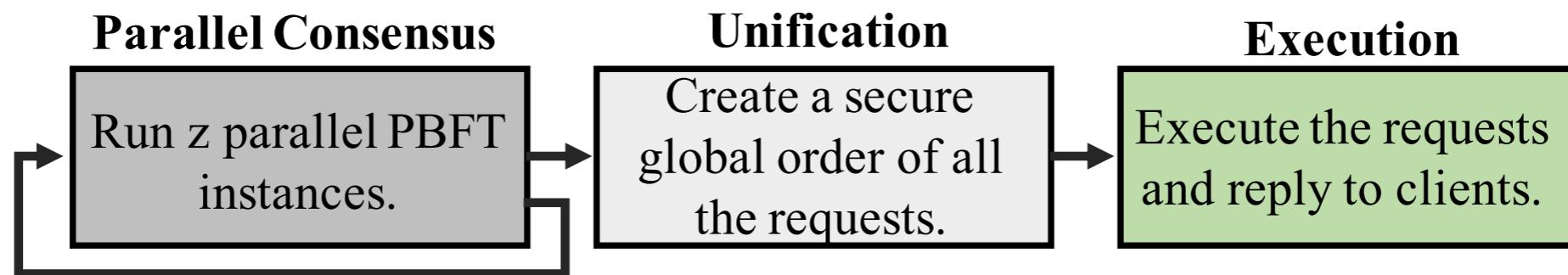


MultiBFT: Scaling Blockchain Databases Through Parallel Resilient Consensus Paradigm [arXiv'19]

A wait-free meta-protocol...

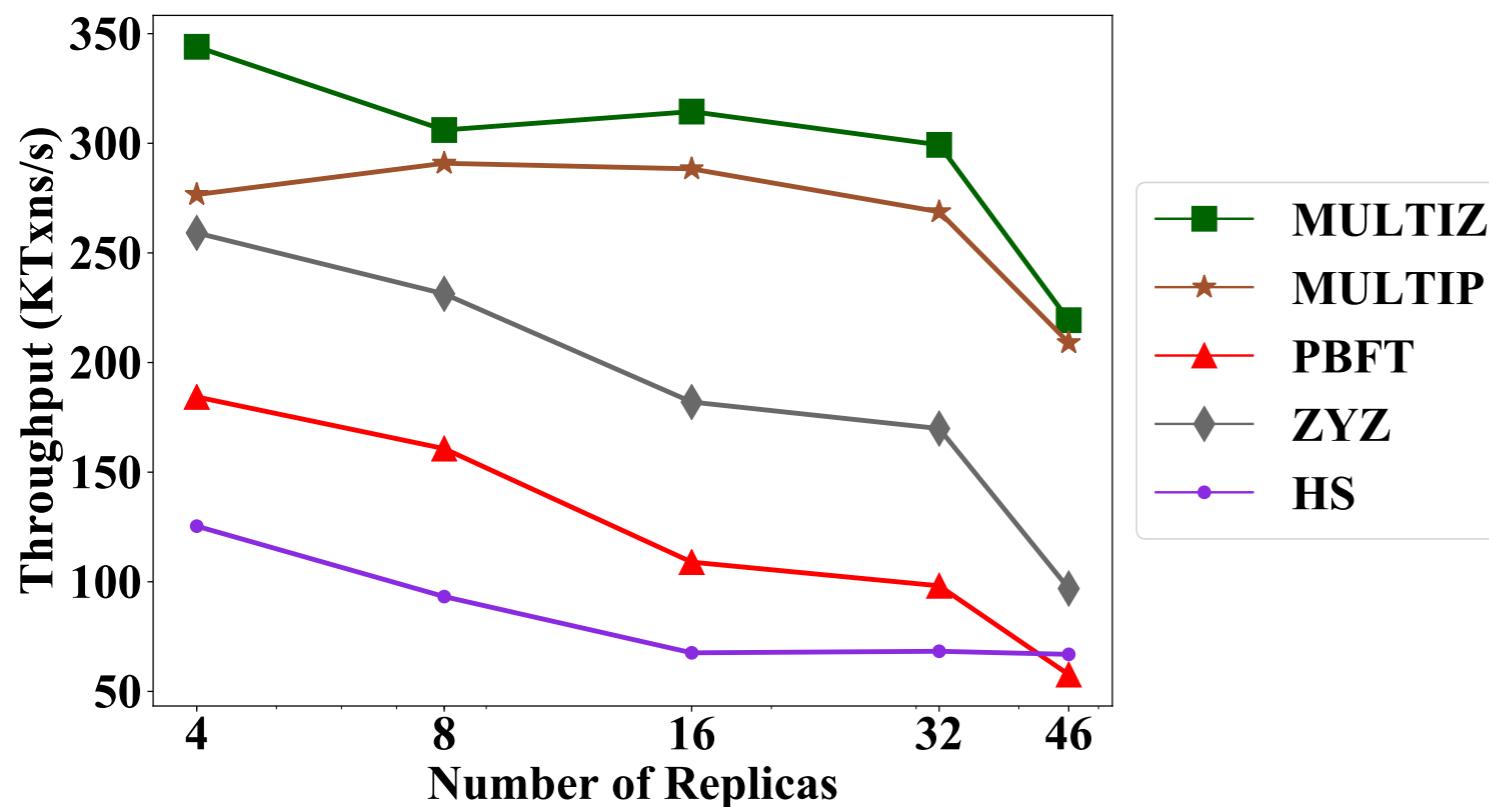
Designate multiple replicas as Primaries!

Run multiple parallel consensuses on each replica independently



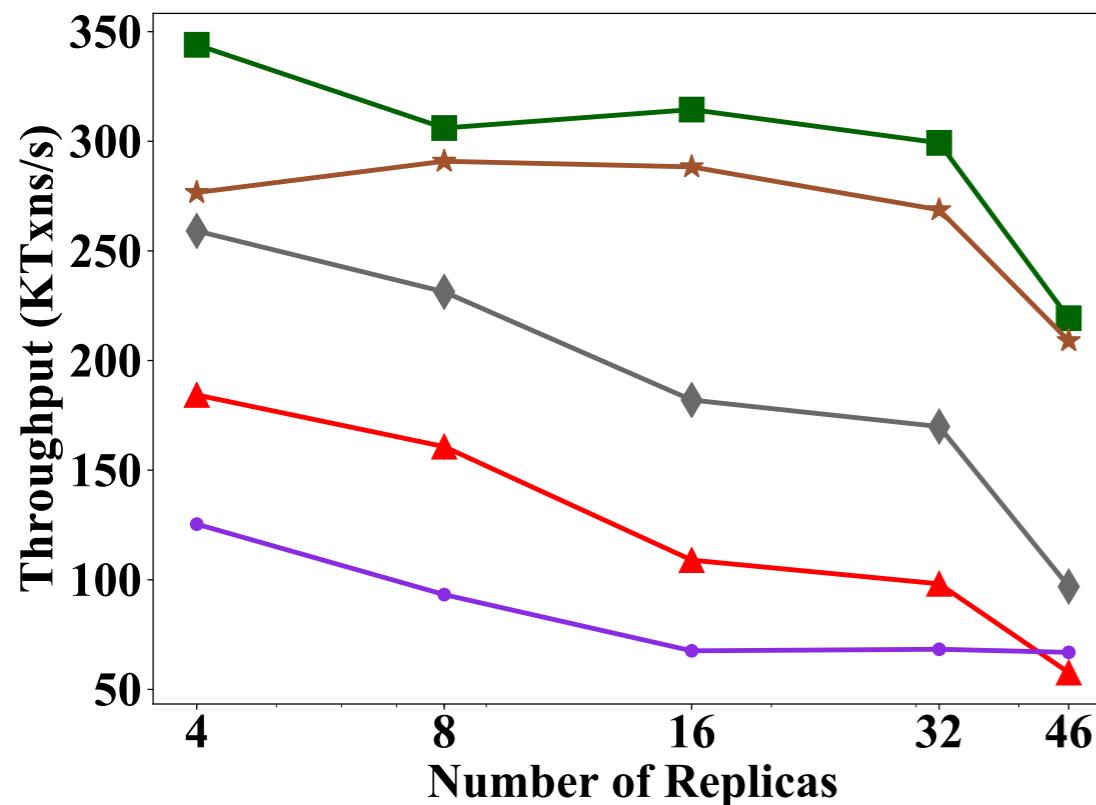
Fault-tolerant MultiBFT Protocol

MultiBFT: Scaling Blockchain Databases Through Parallel Resilient Consensus Paradigm

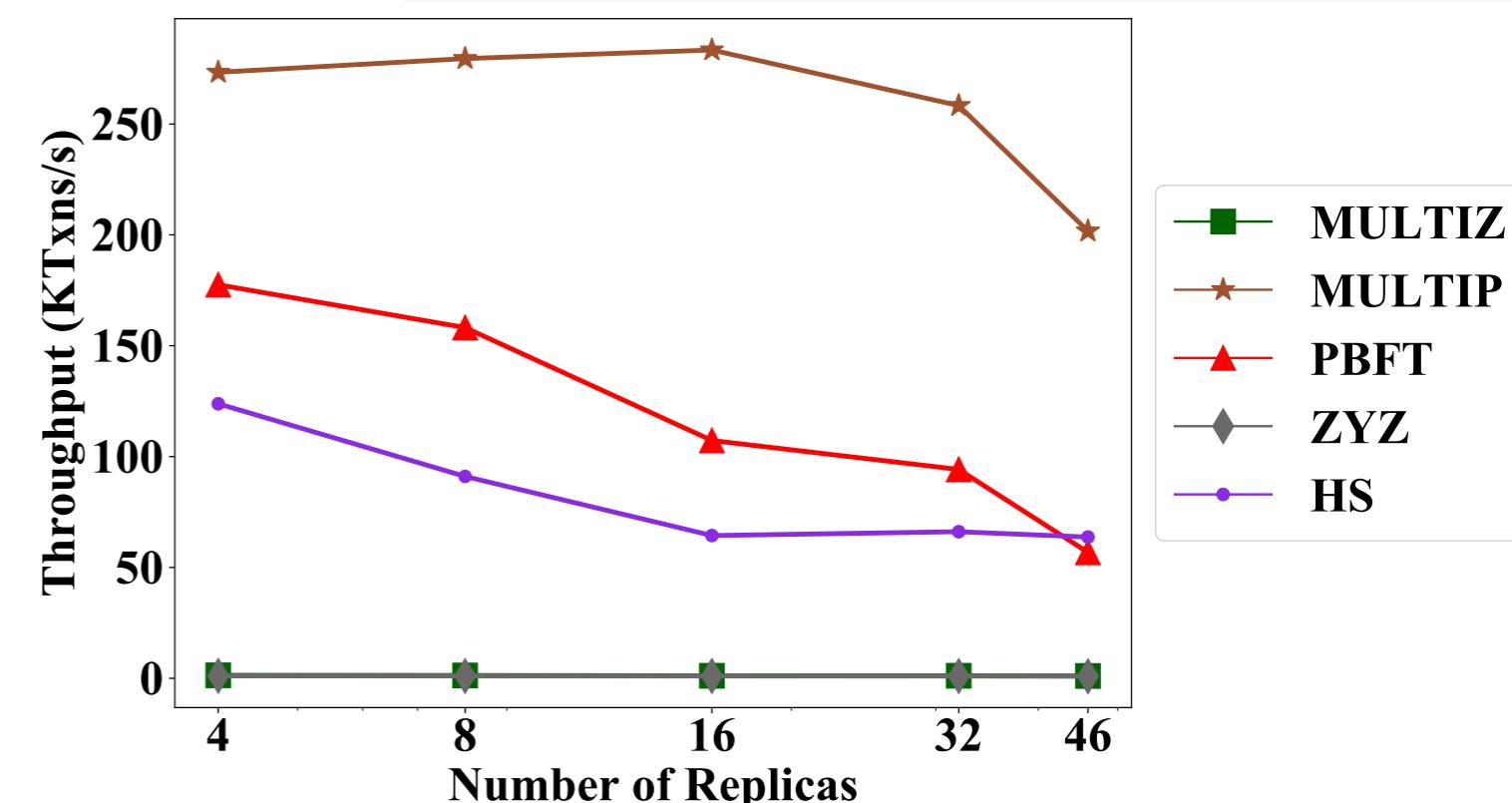


Throughput up to 350,000 txns/s
(without failures)

MultiBFT: Scaling Blockchain Databases Through Parallel Resilient Consensus Paradigm



Throughput up to 350,000 txns/s
(without failures)



Throughput up to 300,000 txns/s
(with failures)

GeoBFT: Global Scale Resilient Blockchain Fabric [VLDB'20]

A meta-protocol, locally running any BFT in parallel and independently

Global ordering provably requires only linear communication

Provably sufficient for primary to send a certificate to at most $f+1$ replicas.

GeoBFT: Global Scale Resilient Blockchain Fabric [VLDB'20]

A meta-protocol, locally running any BFT in parallel and independently
Global ordering provably requires only linear communication

Provably sufficient for primary to send a certificate to at most $\frac{1}{2}n$ replicas.

GeoBFT: Global Scale Resilient Blockchain Fabric [VLDB'20]

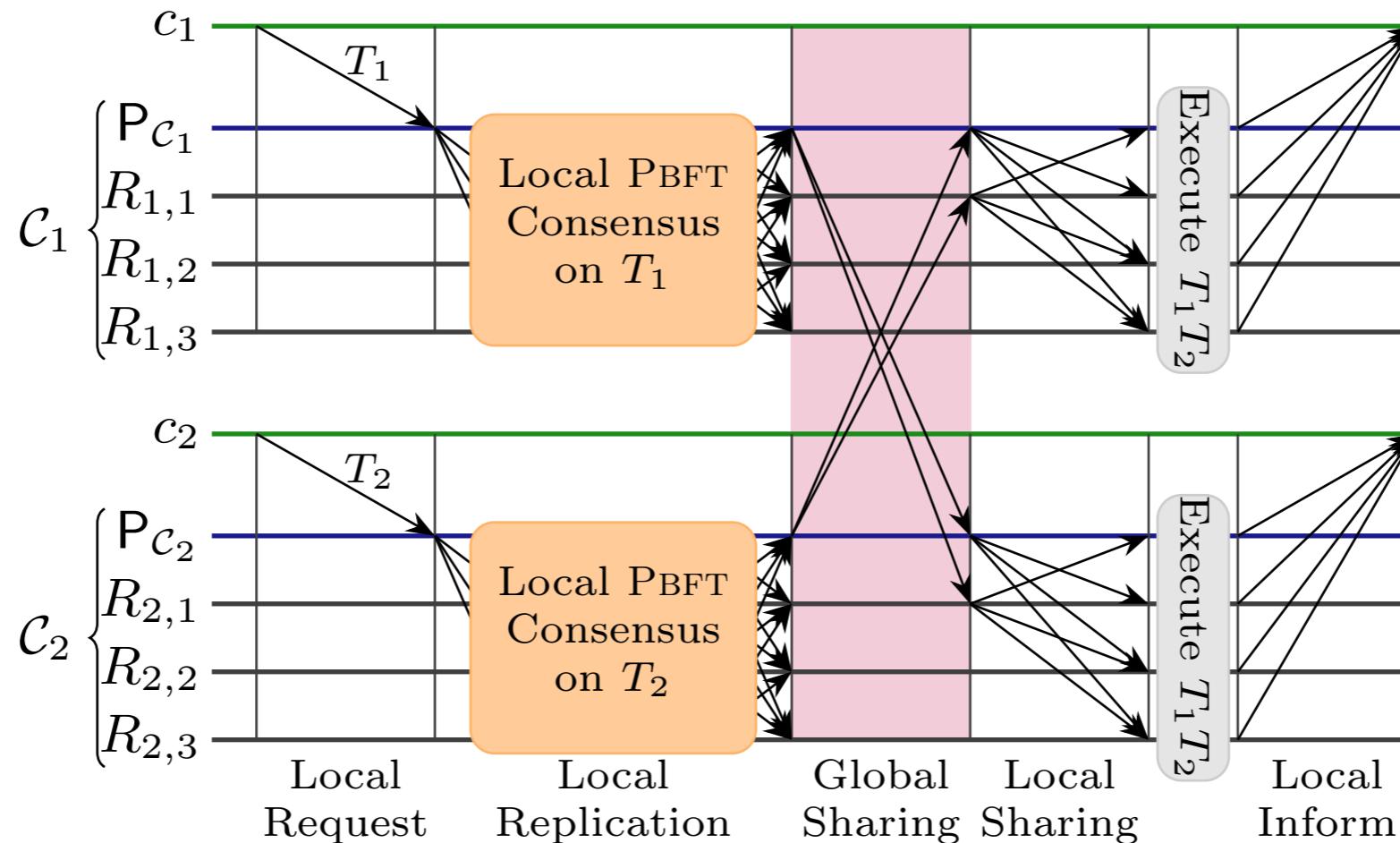
A meta-protocol, locally running any BFT in parallel and independently
Global ordering provably requires only linear communication

Provably sufficient for primary to send a certificate to at most $f+1$ replicas,
malicious primary is detectable and replaceable

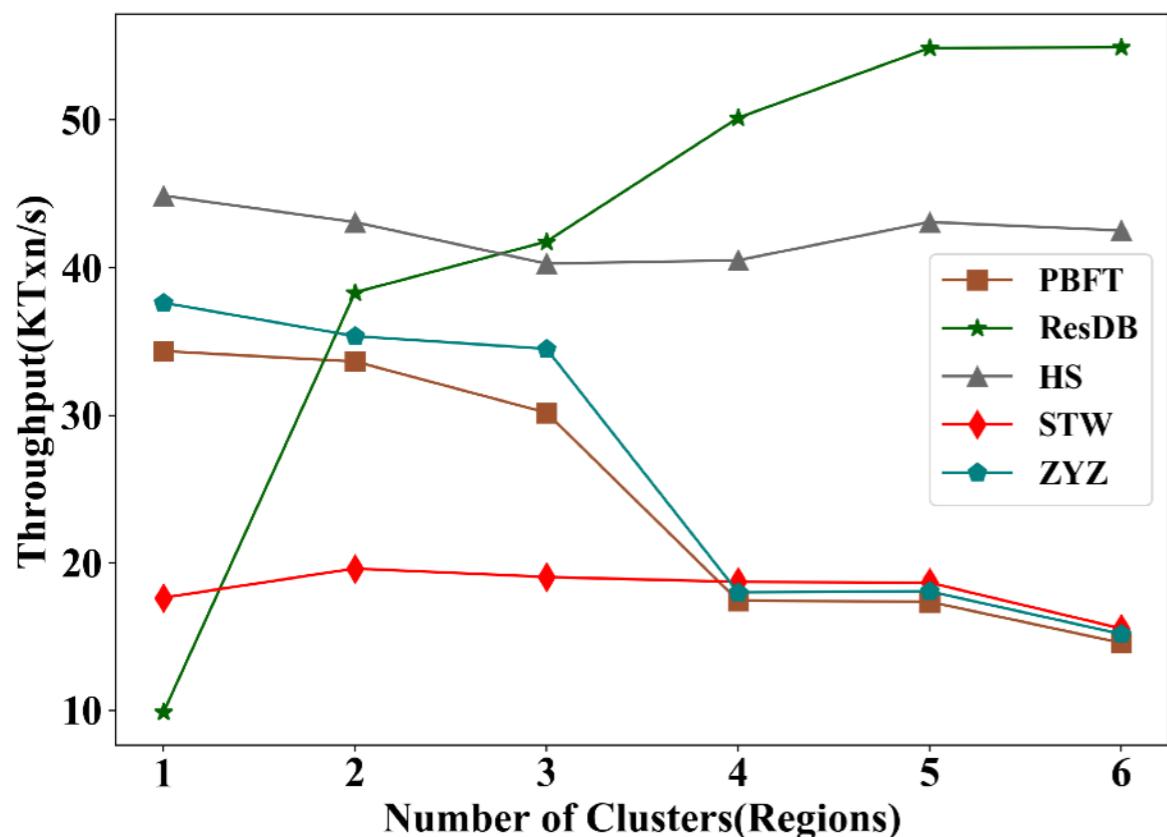
GeoBFT: Global Scale Resilient Blockchain Fabric [VLDB'20]

A meta-protocol, locally running any BFT in parallel and independently
Global ordering provably requires only linear communication

Provably sufficient for primary to send a certificate to at most $f+1$ replicas,
malicious primary is detectable and replaceable

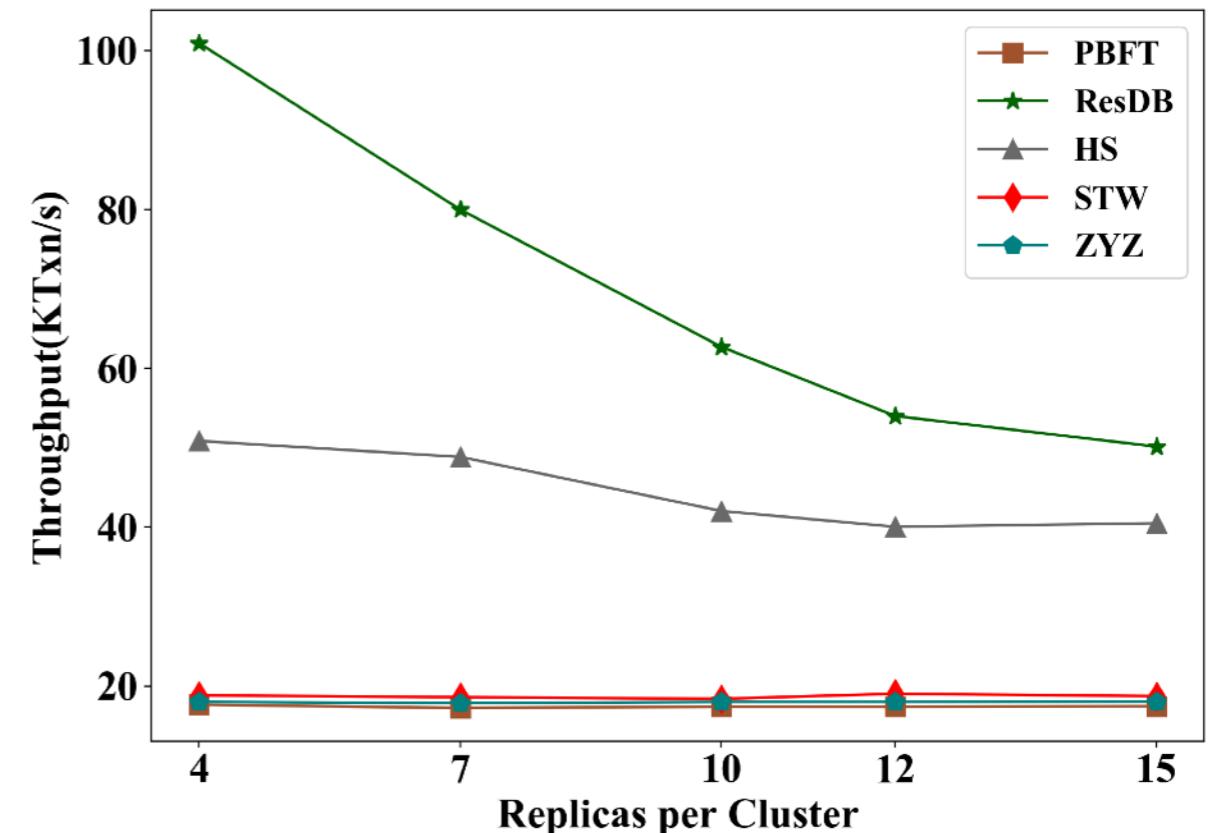
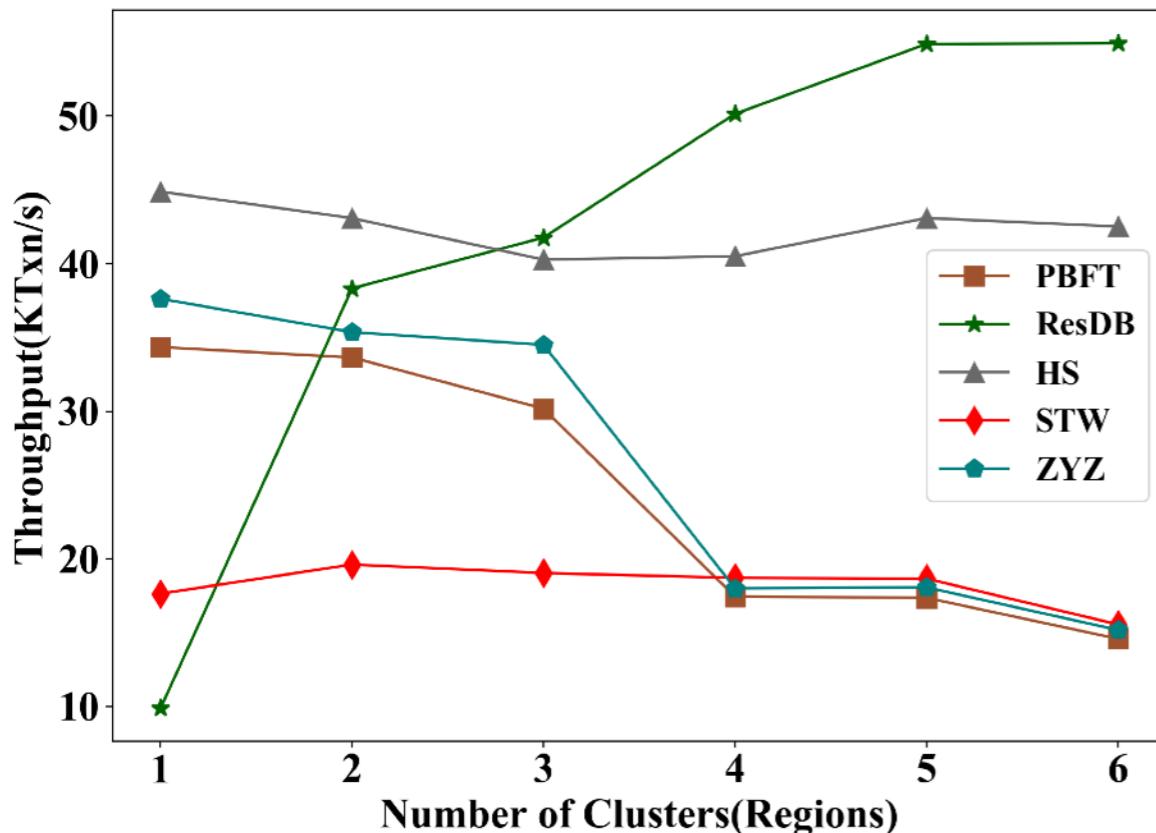


GeoBFT: Global Scale Resilient Blockchain Fabric [VLDB'20]



ResilientDB easily scales across 6 countries in
4 continents due to GeoBFT protocol.

GeoBFT: Global Scale Resilient Blockchain Fabric [VLDB'20]



ResilientDB easily scales across 6 countries in 4 continents due to GeoBFT protocol.

GeoBFT scales a permissioned blockchain up to 60 replicas globally.

The Fault-Tolerant Cluster-Sending Problem [DISC'19]

formalizing the problem of sending a message from one Byzantine cluster to another Byzantine cluster in a reliable manner,

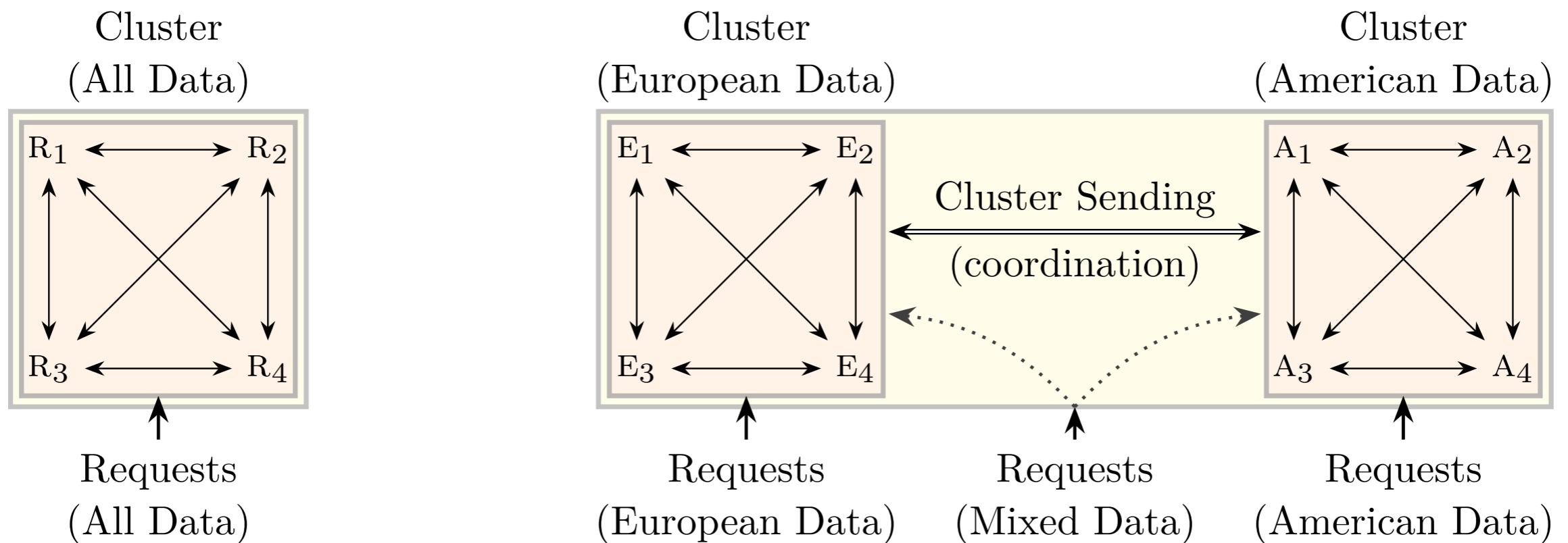
establishing lower bounds on the complexity of this problem under crash failures and Byzantine failures

The Fault-Tolerant Cluster-Sending Problem [DISC'19]

formalizing the problem of sending a message from one Byzantine cluster to another Byzantine cluster in a reliable manner,
establishing lower bounds on the complexity
of this problem under crash failures and Byzantine failures

The Fault-Tolerant Cluster-Sending Problem [DISC'19]

formalizing the problem of sending a message from one Byzantine cluster to another Byzantine cluster in a reliable manner,
establishing lower bounds on the complexity
of this problem under crash failures and Byzantine failures
(linear in the size of clusters)

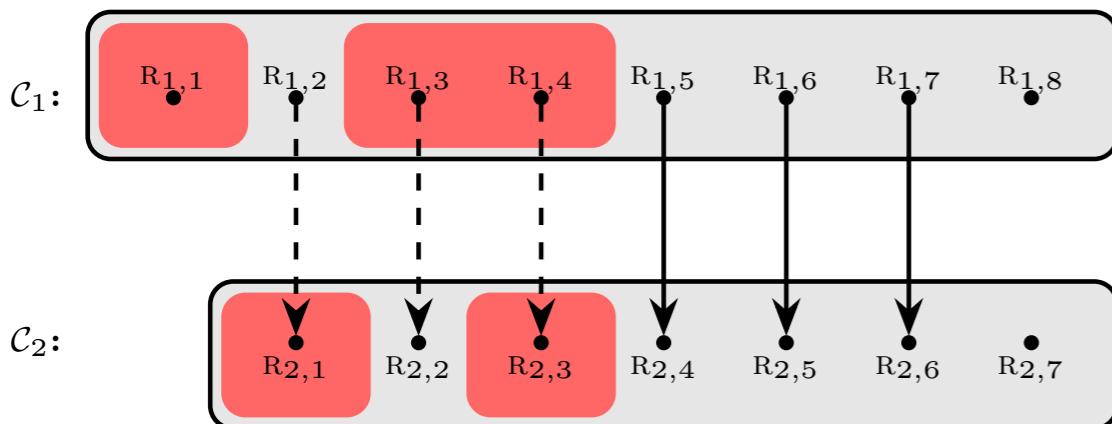


The Fault-Tolerant Cluster-Sending Problem [DISC'19]

formalizing the problem of sending a message from one Byzantine cluster to another Byzantine cluster in a reliable manner,

establishing lower bounds on the complexity of this problem under crash failures and Byzantine failures

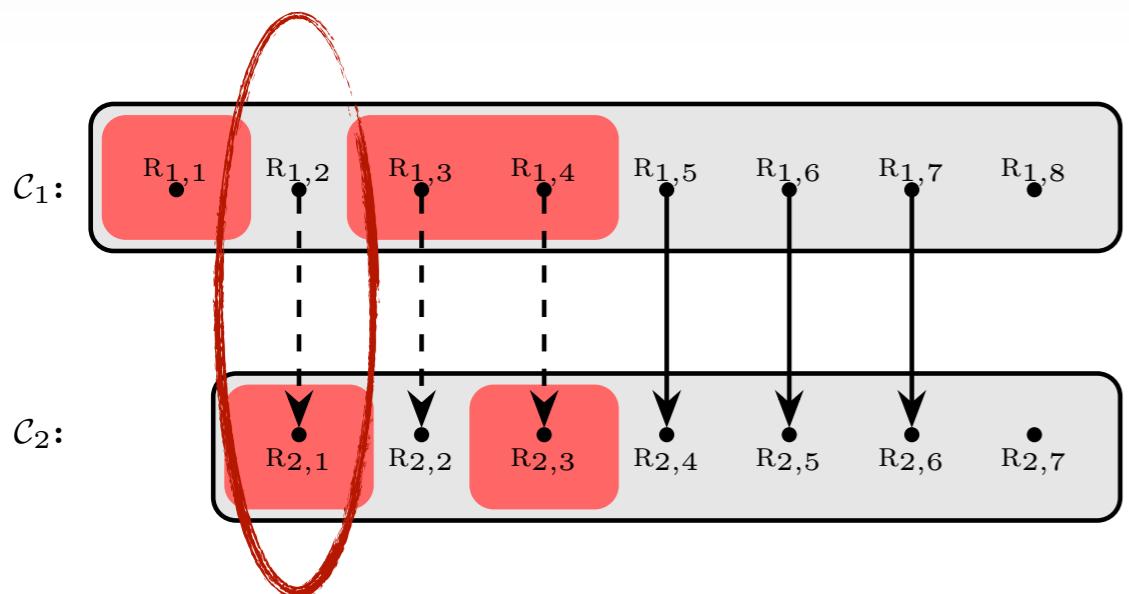
(linear in the size of clusters)



	Protocol	System	Robustness	Messages	Message size
non-linear	RB-bcs	Omit	$n_{C_1} > 2f_{C_1}$, $n_{C_2} > f_{C_2}$	$(f_{C_1} + 1) \cdot (f_{C_2} + 1)$	$\mathcal{O}(\ v\)$
	RB-brs	Byzantine, RS	$n_{C_1} > 2f_{C_1}$, $n_{C_2} > f_{C_2}$	$(2f_{C_1} + 1) \cdot (f_{C_2} + 1)$	$\mathcal{O}(\ v\)$
	RB-bcs	Byzantine, RS	$n_{C_1} > 2f_{C_1}$, $n_{C_2} > f_{C_2}$	$(f_{C_1} + 1) \cdot (f_{C_2} + 1)$	$\mathcal{O}(\ v\ + f_{C_1})$
	RB-bcs	Byzantine, CS	$n_{C_1} > 2f_{C_1}$, $n_{C_2} > f_{C_2}$	$(f_{C_1} + 1) \cdot (f_{C_2} + 1)$	$\mathcal{O}(\ v\)$
linear	PBS-bcs	Omit	$n_{C_1} > 3f_{C_1}$, $n_{C_2} > 3f_{C_2}$	$\mathcal{O}(\max(n_{C_1}, n_{C_2}))$ (optimal)	$\mathcal{O}(\ v\)$
	PBS-brs	Byzantine, RS	$n_{C_1} > 4f_{C_1}$, $n_{C_2} > 4f_{C_2}$	$\mathcal{O}(\max(n_{C_1}, n_{C_2}))$ (optimal)	$\mathcal{O}(\ v\)$
	PBS-bcs	Byzantine, RS	$n_{C_1} > 3f_{C_1}$, $n_{C_2} > 3f_{C_2}$	$\mathcal{O}(\max(n_{C_1}, n_{C_2}))$	$\mathcal{O}(\ v\ + f_{C_1})$
	PBS-bcs	Byzantine, CS	$n_{C_1} > 3f_{C_1}$, $n_{C_2} > 3f_{C_2}$	$\mathcal{O}(\max(n_{C_1}, n_{C_2}))$ (optimal)	$\mathcal{O}(\ v\)$

Byzantine Cluster-Sending in Expected Constant Communication [arXiv'20]

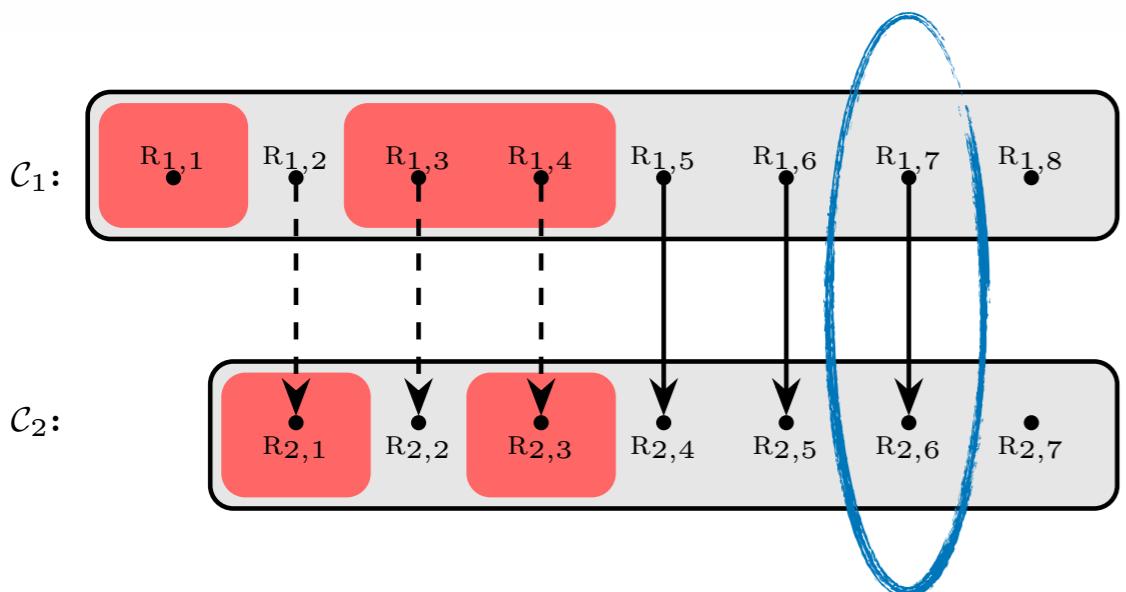
formalizing the problem of probabilistically sending a message from one Byzantine cluster to another Byzantine cluster in a reliable manner,
 establishing lower bounds on the complexity
 of this problem under crash failures and Byzantine failures
 (expected constant message complexity)



	Protocol	Robustness	Message Steps		O.	U.
			(expected)	(worst)		
	PBS-CS [13]	$\min(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2}) > f_{\mathcal{C}_1} + f_{\mathcal{C}_2}$	$f_{\mathcal{C}_1} + f_{\mathcal{C}_2} + 1$	$\max(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2})$	✓	✗
	PBS-CS [13]	$\mathbf{n}_{\mathcal{C}_1} > 3f_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2} > 3f_{\mathcal{C}_2}$			✓	✗
	GEOBFT [12]	$\mathbf{n}_{\mathcal{C}_1} = \mathbf{n}_{\mathcal{C}_2} > 3 \max(f_{\mathcal{C}_1}, f_{\mathcal{C}_2})$	$f_{\mathcal{C}_2} + 1^{\ddagger}$	$\Omega(f_{\mathcal{C}_1} n_{\mathcal{C}_2})$	✗	✓
This Paper	PPCS	$\mathbf{n}_{\mathcal{C}_1} > 2f_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2} > 2f_{\mathcal{C}_2}$	4	$(f_{\mathcal{C}_1} + 1)(f_{\mathcal{C}_2} + 1)$	✗	✓
	PPCS	$\mathbf{n}_{\mathcal{C}_1} > 3f_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2} > 3f_{\mathcal{C}_2}$	$2\frac{1}{4}$	$(f_{\mathcal{C}_1} + 1)(f_{\mathcal{C}_2} + 1)$	✗	✓
	PLCS	$\min(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2}) > f_{\mathcal{C}_1} + f_{\mathcal{C}_2}$	4	$f_{\mathcal{C}_1} + f_{\mathcal{C}_2} + 1$	✓	✓
	PLCS	$\min(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2}) > 2(f_{\mathcal{C}_1} + f_{\mathcal{C}_2})$	$2\frac{1}{4}$	$f_{\mathcal{C}_1} + f_{\mathcal{C}_2} + 1$	✓	✓
	PLCS	$\mathbf{n}_{\mathcal{C}_1} > 3f_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2} > 3f_{\mathcal{C}_2}$	3	$\max(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2})$	✓	✓

Byzantine Cluster-Sending in Expected Constant Communication [arXiv'20]

formalizing the problem of probabilistically sending a message from one Byzantine cluster to another Byzantine cluster in a reliable manner,
 establishing lower bounds on the complexity
 of this problem under crash failures and Byzantine failures
 (expected constant message complexity)



	Protocol	Robustness	Message Steps (expected)	Message Steps (worst)	O.	U.
PBS-CS [13]	$\min(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2}) > f_{\mathcal{C}_1} + f_{\mathcal{C}_2}$		$f_{\mathcal{C}_1} + f_{\mathcal{C}_2} + 1$		✓	✗
	$\mathbf{n}_{\mathcal{C}_1} > 3f_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2} > 3f_{\mathcal{C}_2}$		$\max(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2})$		✓	✗
GEOBFT [12]	$\mathbf{n}_{\mathcal{C}_1} = \mathbf{n}_{\mathcal{C}_2} > 3 \max(f_{\mathcal{C}_1}, f_{\mathcal{C}_2})$		$f_{\mathcal{C}_2} + 1^{\ddagger}$	$\Omega(f_{\mathcal{C}_1} n_{\mathcal{C}_2})$	✗	✓
	$\mathbf{n}_{\mathcal{C}_1} > 2f_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2} > 2f_{\mathcal{C}_2}$		4	$(f_{\mathcal{C}_1} + 1)(f_{\mathcal{C}_2} + 1)$	✗	✓
	$\mathbf{n}_{\mathcal{C}_1} > 3f_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2} > 3f_{\mathcal{C}_2}$		$2\frac{1}{4}$	$(f_{\mathcal{C}_1} + 1)(f_{\mathcal{C}_2} + 1)$	✗	✓
This Paper	PPCS	$\mathbf{n}_{\mathcal{C}_1} > 2f_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2} > 2f_{\mathcal{C}_2}$	4	$f_{\mathcal{C}_1} + f_{\mathcal{C}_2} + 1$	✓	✓
	PPCS	$\mathbf{n}_{\mathcal{C}_1} > 3f_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2} > 3f_{\mathcal{C}_2}$	$2\frac{1}{4}$	$f_{\mathcal{C}_1} + f_{\mathcal{C}_2} + 1$	✓	✓
	PLCS	$\min(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2}) > f_{\mathcal{C}_1} + f_{\mathcal{C}_2}$	4	$f_{\mathcal{C}_1} + f_{\mathcal{C}_2} + 1$	✓	✓
This Paper	PLCS	$\min(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2}) > 2(f_{\mathcal{C}_1} + f_{\mathcal{C}_2})$	$2\frac{1}{4}$	$f_{\mathcal{C}_1} + f_{\mathcal{C}_2} + 1$	✓	✓
	PLCS	$\mathbf{n}_{\mathcal{C}_1} > 3f_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2} > 3f_{\mathcal{C}_2}$	3	$\max(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2})$	✓	✓

Coordination-Free Byzantine Replication With Minimal Communication Costs [ICDT'20]

formalizing the Byzantine learner problem to support efficient analytics for blockchain applications

introducing the delayed-replication algorithm,
utilizing information-dispersal techniques,

Coordination-Free Byzantine Replication With Minimal Communication Costs [ICDT'20]

formalizing the Byzantine learner problem to support efficient

analytics for blockchain applications

introducing the delayed-replication algorithm,

utilizing information dispersal techniques,

giving rise to a coordination-free, push-based, minimal communication protocol

Coordination-Free Byzantine Replication With Minimal Communication Costs [ICDT'20]

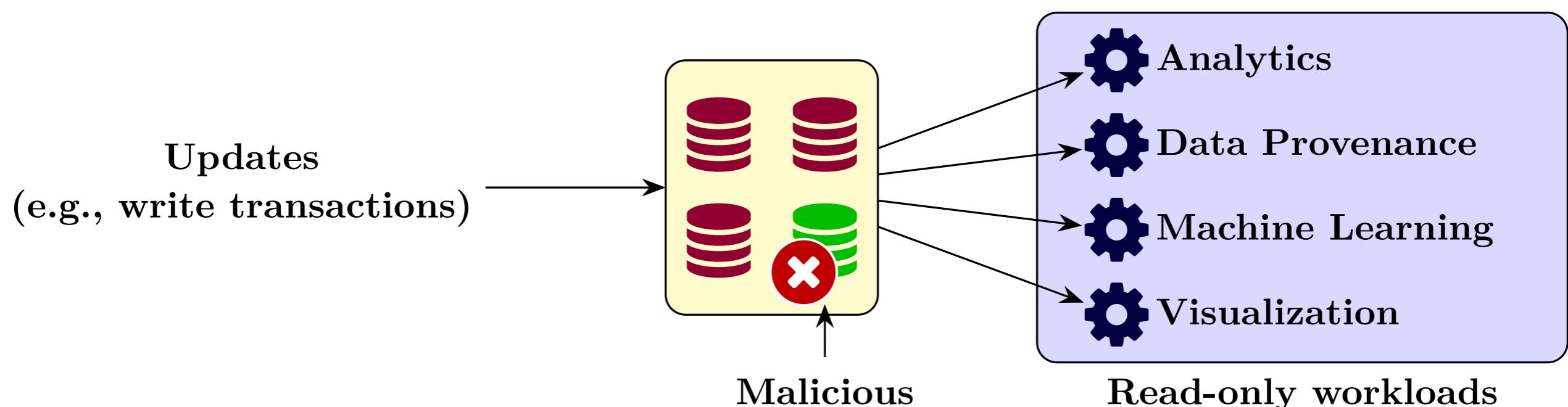
formalizing the Byzantine learner problem to support efficient

analytics for blockchain applications

introducing the delayed-replication algorithm,

utilizing information dispersal techniques,

giving rise to a coordination-free, push-based, minimal communication protocol



Coordination-Free Byzantine Replication With Minimal Communication Costs [ICDT'20]

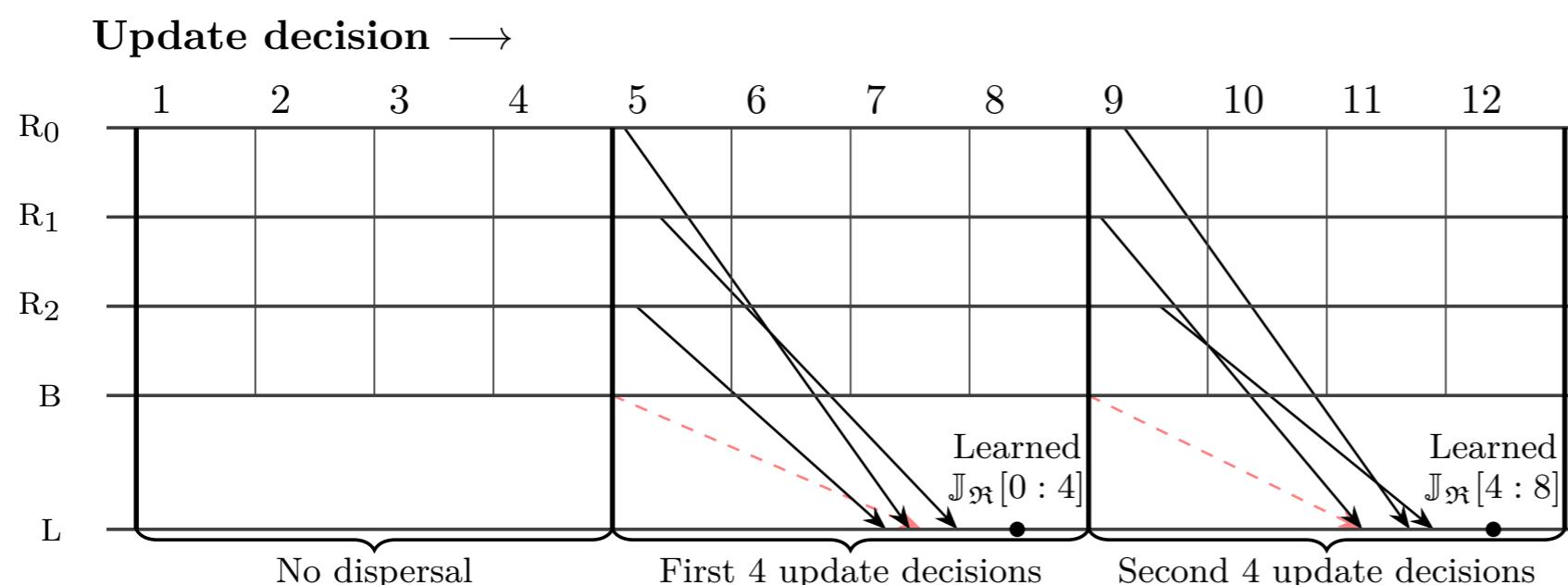
formalizing the Byzantine learner problem to support efficient

analytics for blockchain applications

introducing the delayed-replication algorithm,

utilizing information dispersal techniques,

giving rise to a coordination-free, push-based, minimal communication protocol



System	Checksum	Complexity for the learner		
		Data sent per replica	Data received	Decode steps
$b = 0$	None	$\mathcal{O}(s/g)$	$\mathcal{O}(s(n/g))$	u/n
$b < g$	Simple	$\mathcal{O}(s/g)$	$\mathcal{O}(s(n/g))$	$\binom{g+b}{g}(u/n)$
$b < g$	Tree	$\mathcal{O}(s/g + (u/n) \log(n))$	$\mathcal{O}(s(n/g) + u \log(n))$	u/n

Permissioned Blockchain Through the Looking Glass: Architectural and Implementation Lessons Learned [ICDCS'20]

Single-threaded Monolithic Design

Out-of-ordering Consensus Communication

De-coupled Ordering and Execution

Off-Chain Memory Management

Permissioned Blockchain Through the Looking Glass: Architectural and Implementation Lessons Learned [ICDCS'20]

Single-threaded Monolithic Design

Out-of-ordering Consensus Communication

De-coupled Ordering and Execution

Off-chain Memory Management

Permissioned Blockchain Through the Looking Glass: Architectural and Implementation Lessons Learned [ICDCS'20]

Single-threaded Monolithic Design

Out-of-ordering Consensus Communication

De-coupled Ordering and Execution

→ Parallel Memory Management

Permissioned Blockchain Through the Looking Glass: Architectural and Implementation Lessons Learned [ICDCS'20]

Single-threaded Monolithic Design

Out-of-ordering Consensus Communication

De-coupled Ordering and Execution

Off-Chain Memory Management

Permissioned Blockchain Through the Looking Glass: Architectural and Implementation Lessons Learned [ICDCS'20]

Single-threaded Monolithic Design

Out-of-ordering Consensus Communication

De-coupled Ordering and Execution

Off-Chain Memory Management

Expensive Cryptographic Practices (DS vs. MAC)

Permissioned Blockchain Through the Looking Glass: Architectural and Implementation Lessons Learned [ICDCS'20]

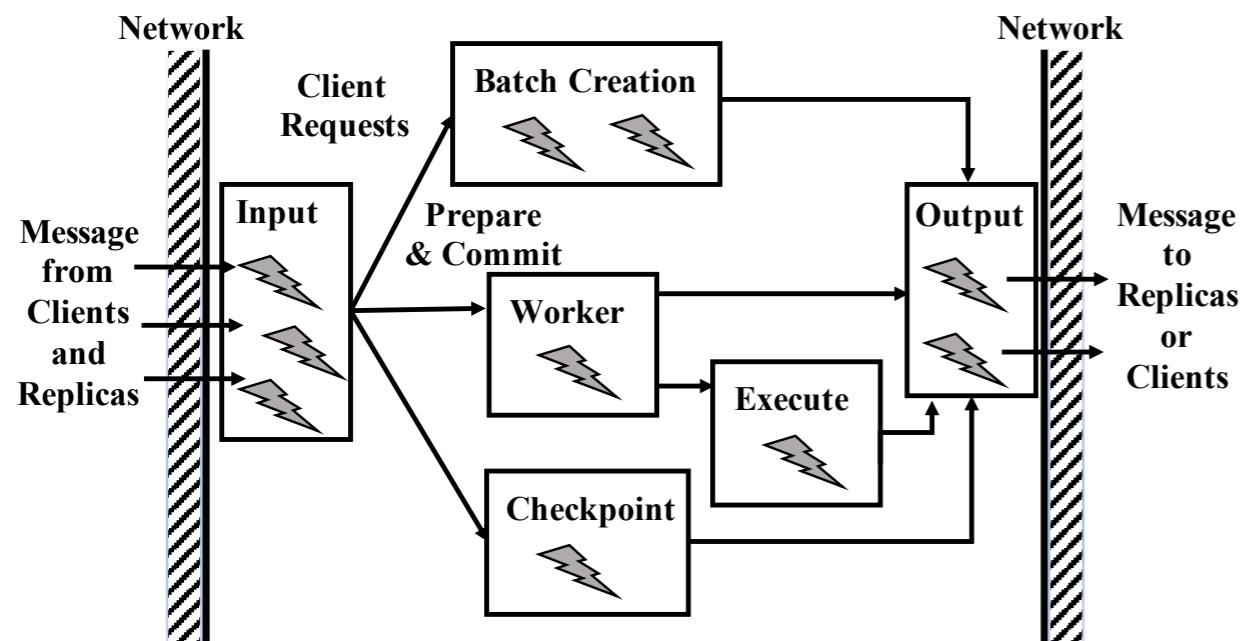
Single-threaded Monolithic Design

Out-of-ordering Consensus Communication

De-coupled Ordering and Execution

Off-Chain Memory Management

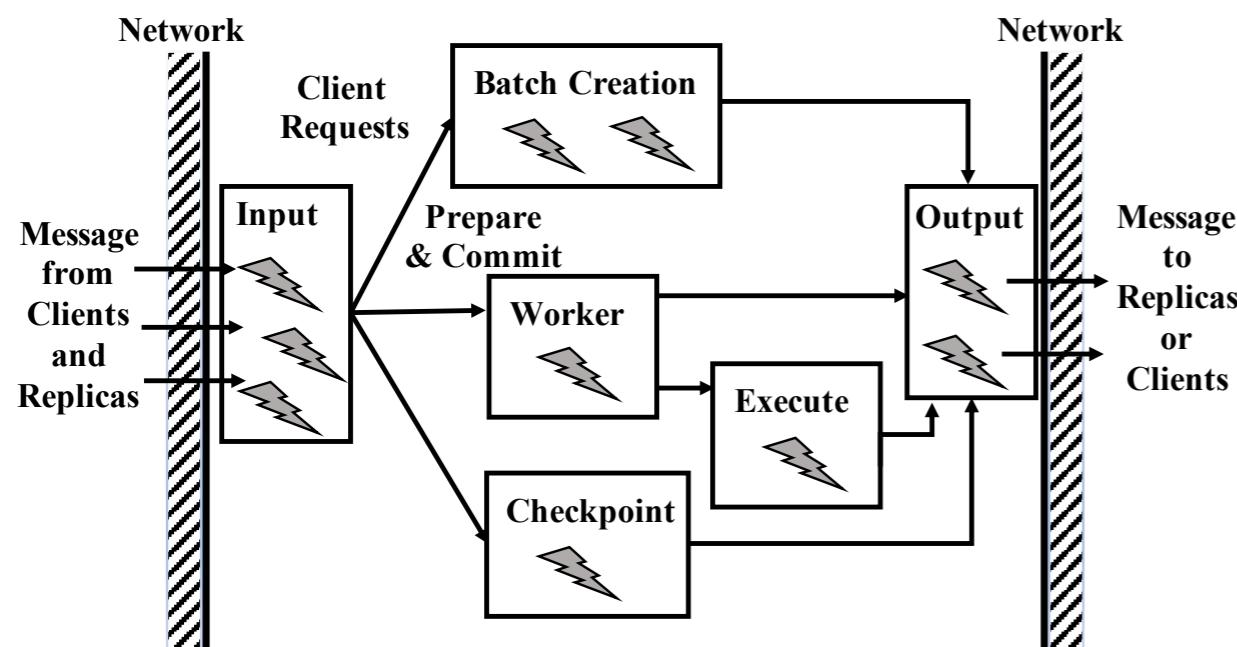
Expensive Cryptographic Practices (DS vs. MAC)



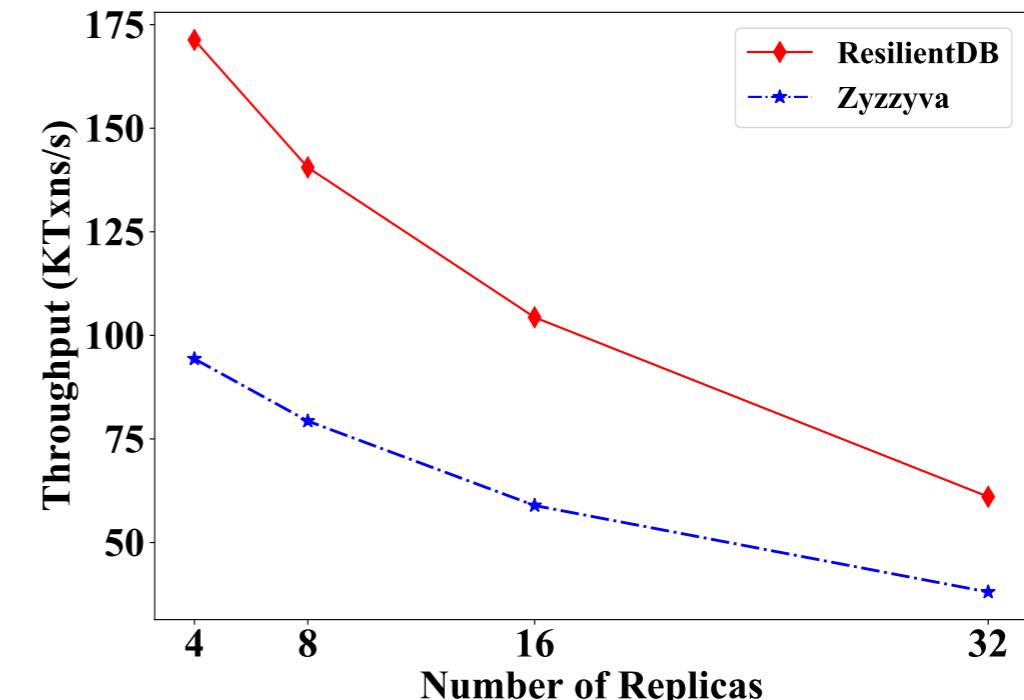
Multi-Threaded Deep Pipeline

Permissioned Blockchain Through the Looking Glass: Architectural and Implementation Lessons Learned [ICDCS'20]

Single-threaded Monolithic Design
Out-of-ordering Consensus Communication
De-coupled Ordering and Execution
Off-Chain Memory Management
Expensive Cryptographic Practices (DS vs. MAC)



Multi-Threaded Deep Pipeline



Can a well-crafted system based on a classical BFT protocol outperform a modern protocol?

Revisit Resiliency

(Graduate Student Experiment Continues)

Mount Tallac, Lake Tahoe

12.1 Miles Long

**3,931 Feet Elevation Gain
(9,738 Feet at Summit)**

Mount Tallac, Lake Tahoe
12.1 Miles Long
3,931 Feet Elevation Gain
(9,738 Feet at Summit)



Mount Tallac, Lake Tahoe
12.1 Miles Long
3,931 Feet Elevation Gain
(9,738 Feet at Summit)



Fostering Resiliency

(Offering Stress Management and Well-Being Courses at UC Davis)



*Release Tension
Increase Focus*

Days: Wednesdays

Time: 7:00 pm - 8:00 pm

Location: Zoom (Live Online Class)

INSTRUCTORS:

Mohammad Sadoghi, Ph.D.
Nasim Bahadorani, DrPH.

Computer Science Department
UCDAVIS
UNIVERSITY OF CALIFORNIA

Dress Code:
Loose comfortable clothing,
sweat shirts and pants with socks.



Becoming an EXTRAORDINARY Human

Spring 2020

Days: Thursdays

Time: 7:00 pm - 8:00 pm

Location: Zoom

CRN: 57877

INSTRUCTORS: Mohammad Sadoghi, Ph.D.
Nasim Bahadorani, DrPH.

No one wants to be ordinary. This course focuses on the personal development of the characteristics of human beings deemed extraordinary. Outcomes include enhanced concentration for higher-level cognition, increased capacity to handle stress, development of increased self-confidence, increased mastery of emotional and mental processes, development of physical awareness and control, and development of positive personal characteristics. Physical activities include movements and visualizations.

msadoghi@ucdavis.edu

Computer Science Department
UCDAVIS
UNIVERSITY OF CALIFORNIA

Reduce Stress

**First-year Seminar (FYS):
Undergraduate Survival Kit**

Learn the foundation & working knowledge of stress reduction based on a unique heart-centered meditation method referred to as Tamarkoz®.

The M.T.O. Tamarkoz® method is the art of self-knowledge through concentration and meditation.

Spring 2020

Time: Tuesdays from 7:00pm-8:00pm

Location: UCDAVIS Zoom (CRN: 66553)

mto tamarkoz BE BALANCED

*Release Tension
Increase Focus*

UCDAVIS
UNIVERSITY OF CALIFORNIA

THE CALIFORNIA AGGIE
Seminar spotlight: "Becoming an Extraordinary Human"
The California Aggie, April 6, 2020





THANK YOU

FOR COMPLETE REFERENCES

