Distributed Data Persistency

Apostolos Kokolis, Antonis Psistakis, Benjamin Reidys,
Jian Huang and Josep Torrellas

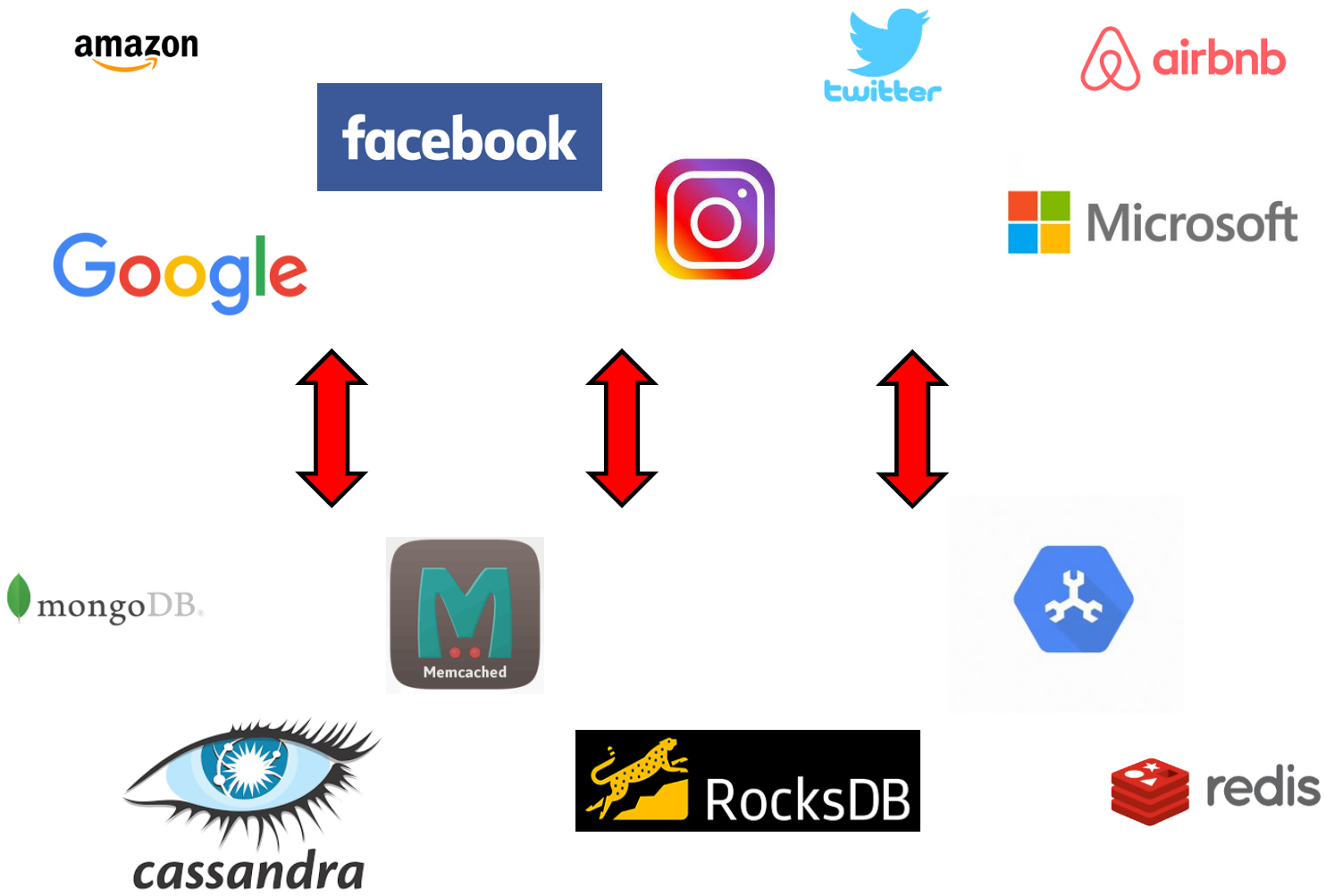
University of Illinois at Urbana-Champaign

International Symposium on Microarchitecture (MICRO) 2021

October 2021

ILLINOIS

Distributed Data-Stores



Persistent Storage

- Disks & Network are slow

- Clients require high-throughput and low latency
- Distributed applications avoid frequent writes to persistent storage
  - Client latency is important

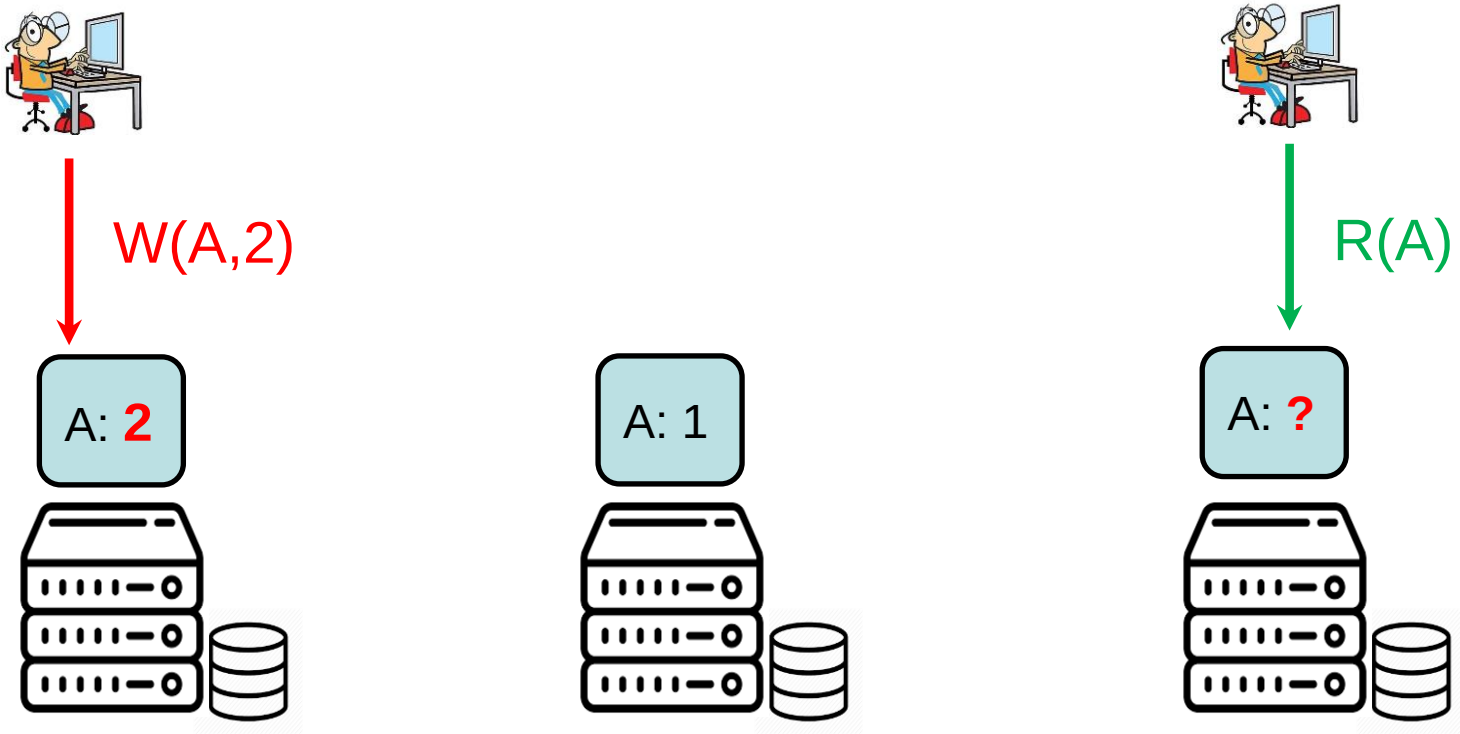
Replicate data and defer persistence

Data Replication in Distributed Systems

- Keep a copy of the data in different machines (replicas)
  - Increases availability and throughput
  - Provides some fault-tolerance

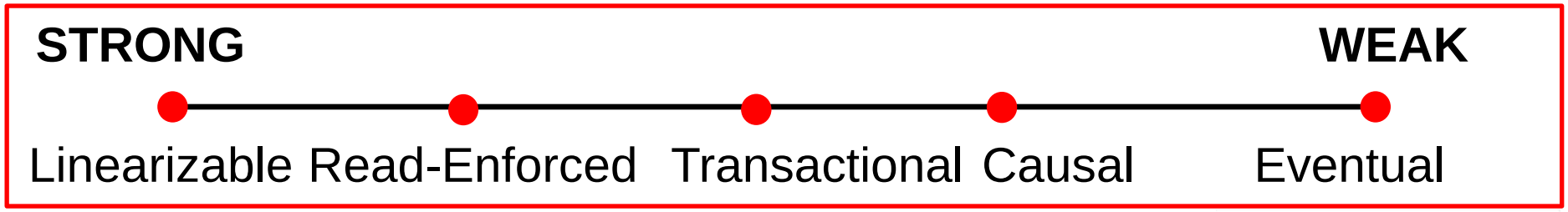
Data Consistency

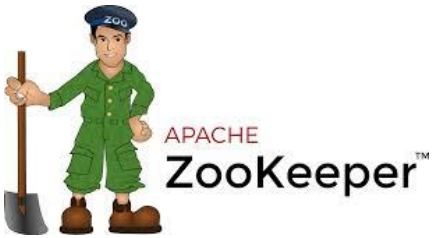
- The challenge is to keep the replicas consistent



Data Consistency Models

- Consistency defines the values that a client can read
- Models range
  - **Strong:** need the values to propagate immediately
  - **Weak:** can have stale data in replicas





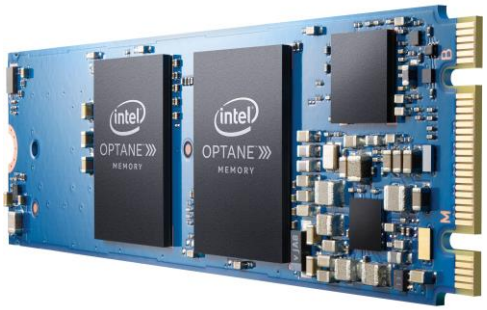
Supporting Data Persistency

- There is scant attention to persistency
- Mostly there are two options
  - Eagerly persist → persist on replication
  - Eventually persist → persist in the background
- Reduced support for durability suffers
  - Data loss from whole system or majority failures
  - Slow recovery



Hardware Advances

- NVM
  - Provides durability in hundreds of ns



- Network
  - RDMA has reached sub-µs latency

Contribution: Distributed Data Persistency (DDP)

- DDP models: integrate consistency models and persistency models in a distributed system
- Create a set of new DDP models
- Design low-latency protocols that support the DDP models
- Provide a trade-off analysis of the DDP models
  - Performance
  - Durability
  - Programmer intuition
  - Programmability
  - Implementability
- Implications of the DDP models on distributed applications

Consistency vs Persistency

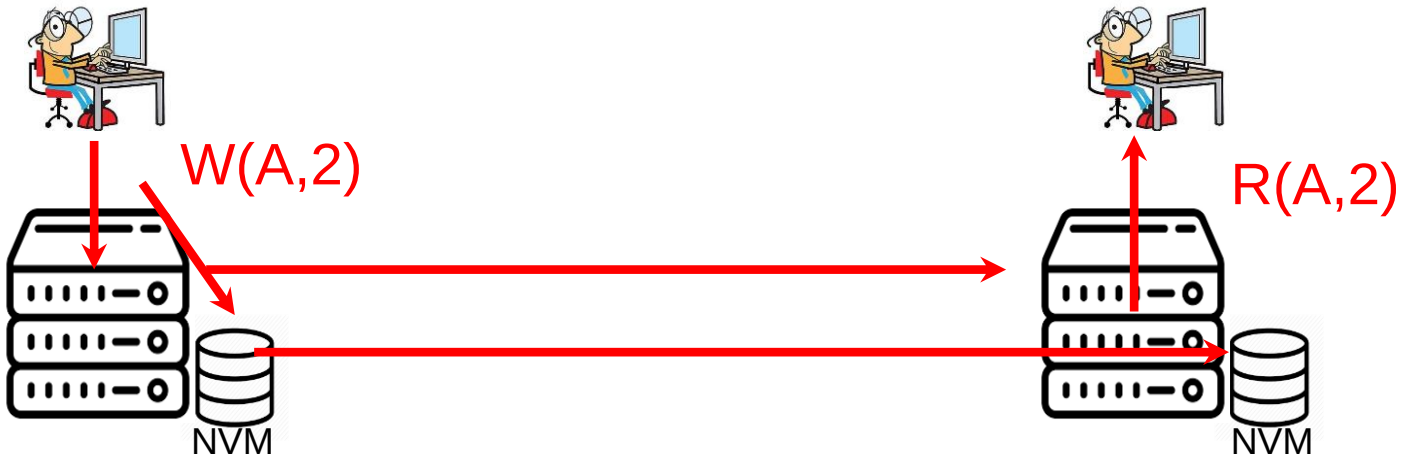
- Consider a distributed system with two replica nodes

Visibility point (VP)

Consistency defines the VP. When an update becomes available for consumption in replica nodes.

Durability point (DP)

Persistency defines the DP. When an update becomes durable and cannot be lost by a failure.



Consistency and Persistency Models

Consistency Model	Visibility Point of Updates

Persistency Model	Durability Point of Updates

Consistency and Persistency Models

Consistency Model	Visibility Point of Updates
Linearizable	When the update occurs
Read-Enforced	Before the update is read
Transactional	At the end of the transaction
Causal	After the VPs of updates in causal history
Eventual	Sometime in the future

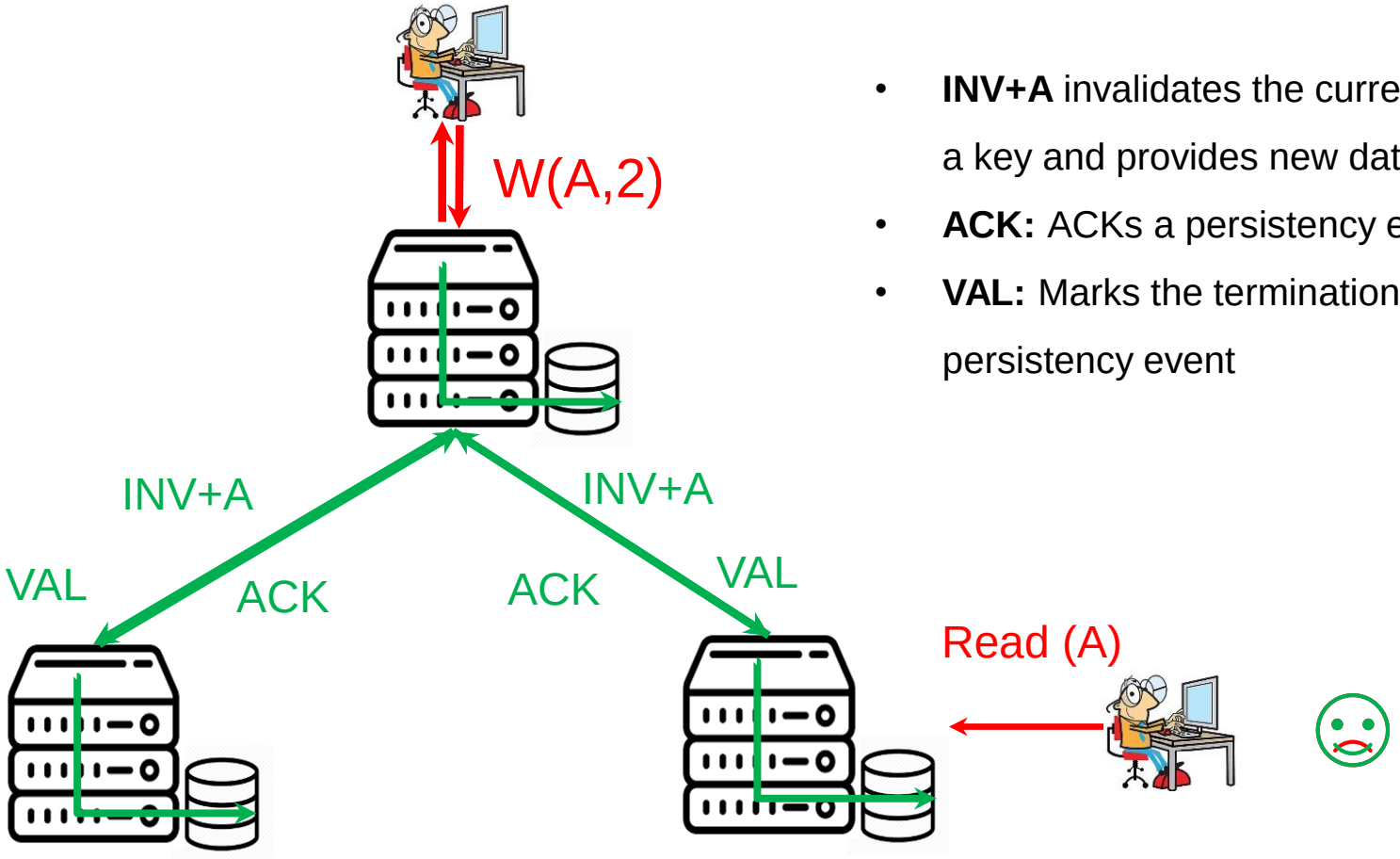
Persistency Model	Durability Point of Updates

Consistency and Persistency Models

Consistency Model	Visibility Point of Updates
Linearizable	When the update occurs
Read-Enforced	Before the update is read
Transactional	At the end of the transaction
Causal	After the VPs of updates in causal history
Eventual	Sometime in the future

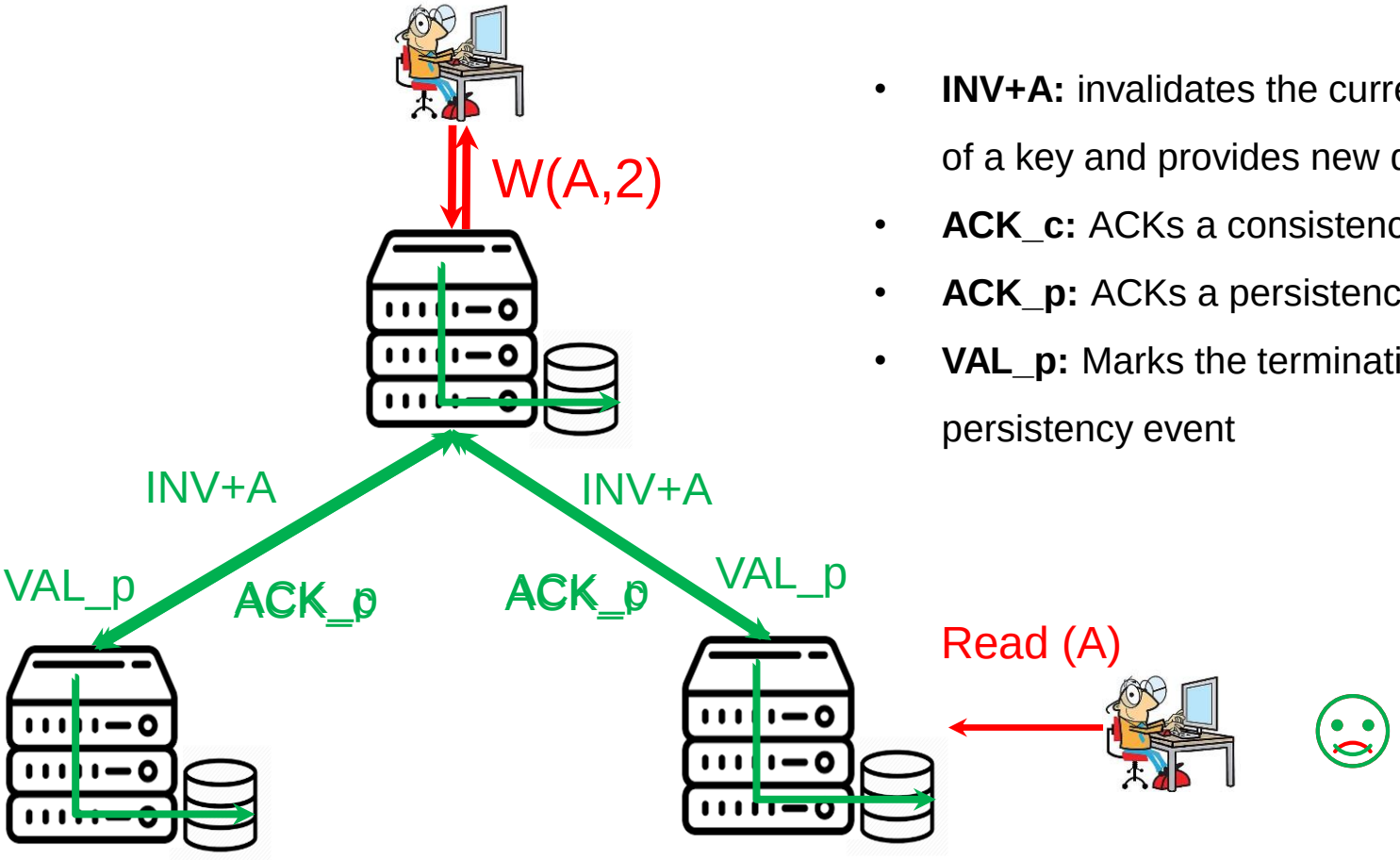
Persistency Model	Durability Point of Updates
Strict	When the update occurs
Synchronous	At the VP of the update
Read-Enforced	Before the update is read
Scope	At the end of scope
Eventual	Sometime in the future

DDP Protocols <Linear, Synchronous>



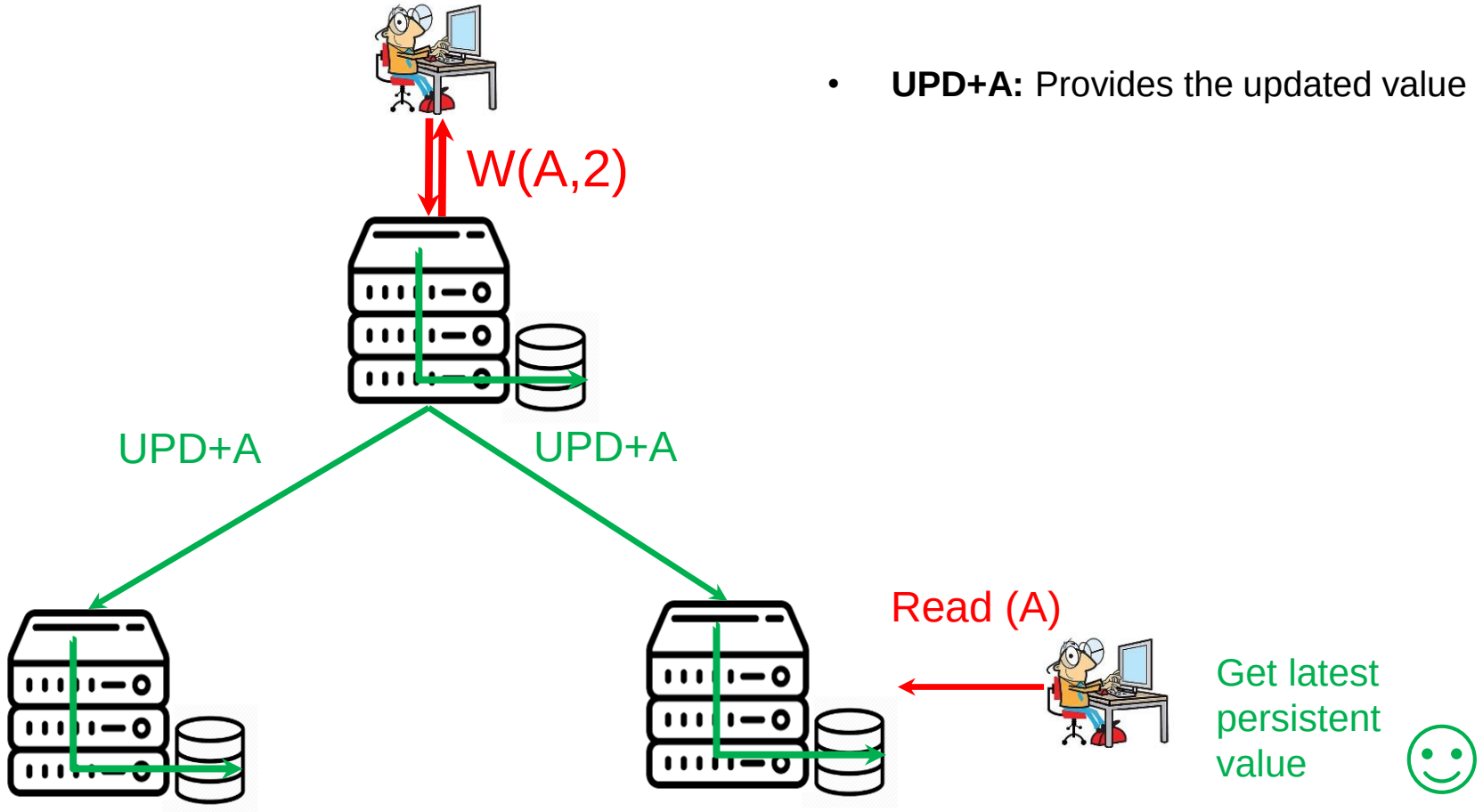
- **INV+A** invalidates the current value of a key and provides new data
- **ACK:** ACKs a persistency event
- **VAL:** Marks the termination of a persistency event

DDP Protocols <Linear, Read-Enforced>



- **INV+A:** invalidates the current value of a key and provides new data
- **ACK_c:** ACKs a consistency event
- **ACK_p:** ACKs a persistency event
- **VAL_p:** Marks the termination of a persistency event

DDP Protocols < Eventual, Synchronous>



- **UPD+A:** Provides the updated value

DDP Tradeoffs

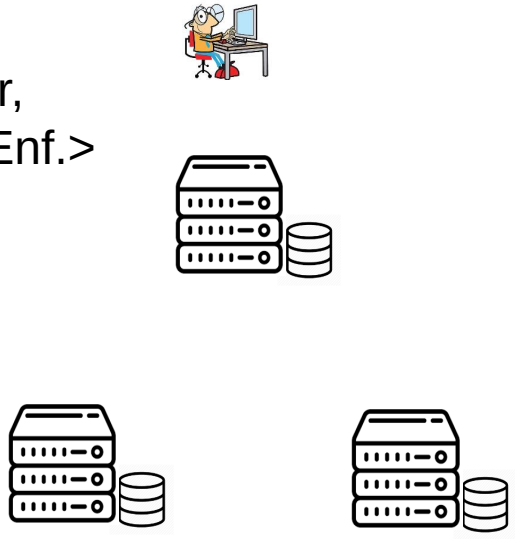
Cons.	Pers.

DDP Tradeoffs

Cons.	Pers.	Dura-bility	Performance				Programmer Intuition		
			Wr Opt?	Rd Opt?	Traffic	Overall	Monotonic Rds?	Non-Stale Rds?	Overall
Linear	Synch.								
Linear	Read-Enf.								

<Linear, Synch.>

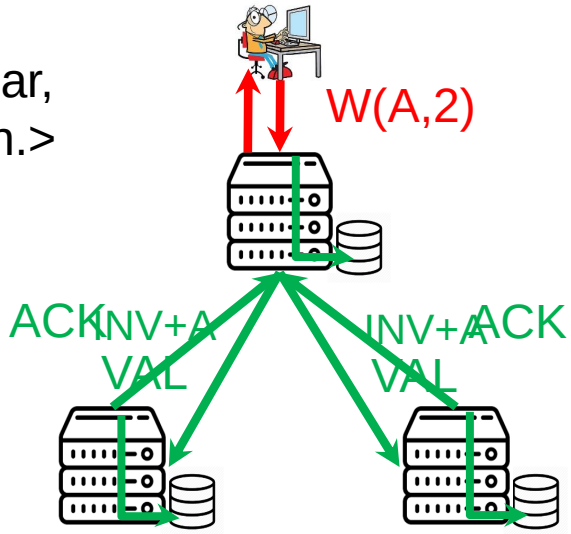
<Linear, Read-Enf.>



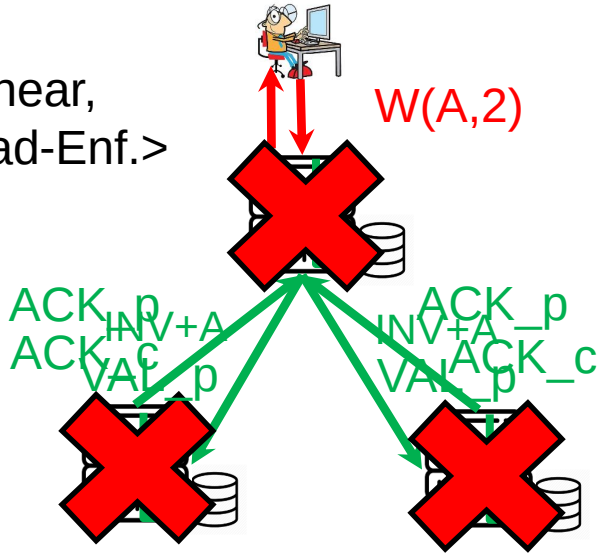
DDP Tradeoffs

Cons.	Pers.	Dura-bility	Performance				Programmer Intuition		
			Wr Opt?	Rd Opt?	Traffic	Overall	Monotonic Rds?	Non-Stale Rds?	Overall
Linear	Synch.	↑	✗	✗	⟷	↓	✓	✓	↑
Linear	Read-Enf.	⟷	✓	✗	↑	⟷	✓	✗	⟷

<Linear, Synch.>



<Linear, Read-Enf.>



DDP Tradeoffs

Cons.	Pers.	Dura-bility	Performance				Programmer Intuition		
			Wr Opt?	Rd Opt?	Traffic	Overall	Monotonic Rds?	Non-Stale Rds?	Overall
Linear	Synch.	↑	✗	✗	↔	↓	✓	✓	↑
Linear	Read-Enf.	↔	✓	✗	↔	↔	✓	✗	↔
Event.	Sync.	↓	✓	✓	↓	↑	✗	✗	↓

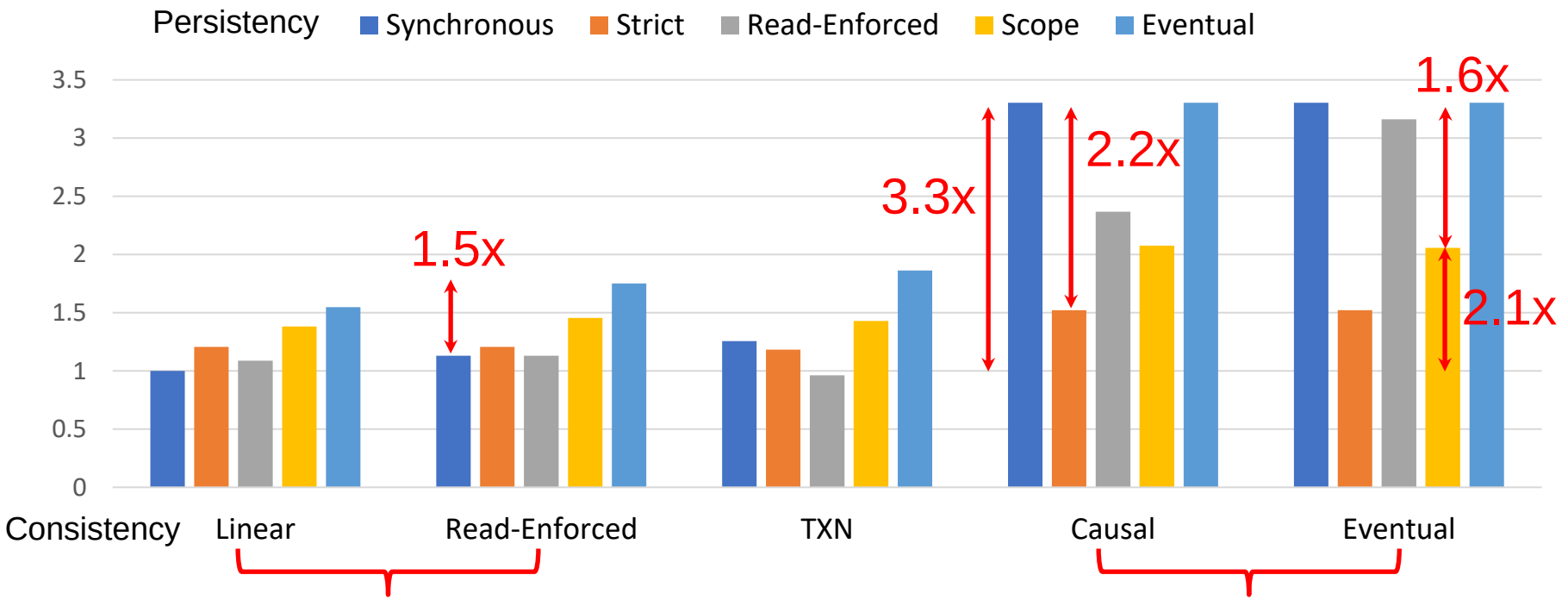
- Weak consistency models sacrifice durability for performance
- Intuition is reduced → updates can be lost

- Transactional consistency and scope persistency have low programmability
- Programmers add TXNs and Scopes

Methodology

- Simulate a system of 5 servers with NVM memory using SST
  - 20-cores per server, 1-client per core
  - Model future RDMA primitives and high-end NICs
  - DDIO to LLC through RDMA
- High performance network
  - 1µs NIC-NIC latency
  - 200Gb/s
- Applications running YCSB workloads
  - Memcached, HashTable, Map, B-Tree, B+ Tree

Evaluation - Throughput



- Models with strong consistency have low throughput
- Causal and Eventual consistency deliver high performance
- For a consistency model, persistency makes a big difference

Implications to Applications

Developers can choose the correct DDP model according to the app

- Latency sensitive that can tolerate staleness
  - Web browsing, social networks
  - **<Eventual, Synchronous>**
    - Low latency, High throughput, Simple programming 
- Consistency sensitive applications
  - Bounded staleness web services, aggregate data from users
  - **<Read-Enf, Scope/Eventual>**
    - High throughput, Low tail latency 
- Geo-distributed systems: hybrid models
  - Within a cluster: **<Linear, Eventual>**
  - Across different clusters: **<Eventual, Synchronous>**
    - Strong consistency within a cluster and strong persistency across clusters 

Conclusions

- Introduced DDP models: integrate consistency models and persistency models in a distributed system
- Created a set of DDP models and low-latency protocols that support them
- Provided a trade-off analysis of the DDP models
- Discussed implications of the DDP models on applications
- Some of the results shown:
  - Models combining strong cons. with weak pers., or weak cons. with strong pers. are often highly competitive
  - Causal cons. combined with different memory pers. models is often good choice

Distributed Data Persistency

Apostolos Kokolis, Antonis Psistakis, Benjamin Reidys,
Jian Huang and Josep Torrellas
{kokolis2,psistaki,breidys2,jianh,torrella}@illinois.edu

University of Illinois at Urbana-Champaign

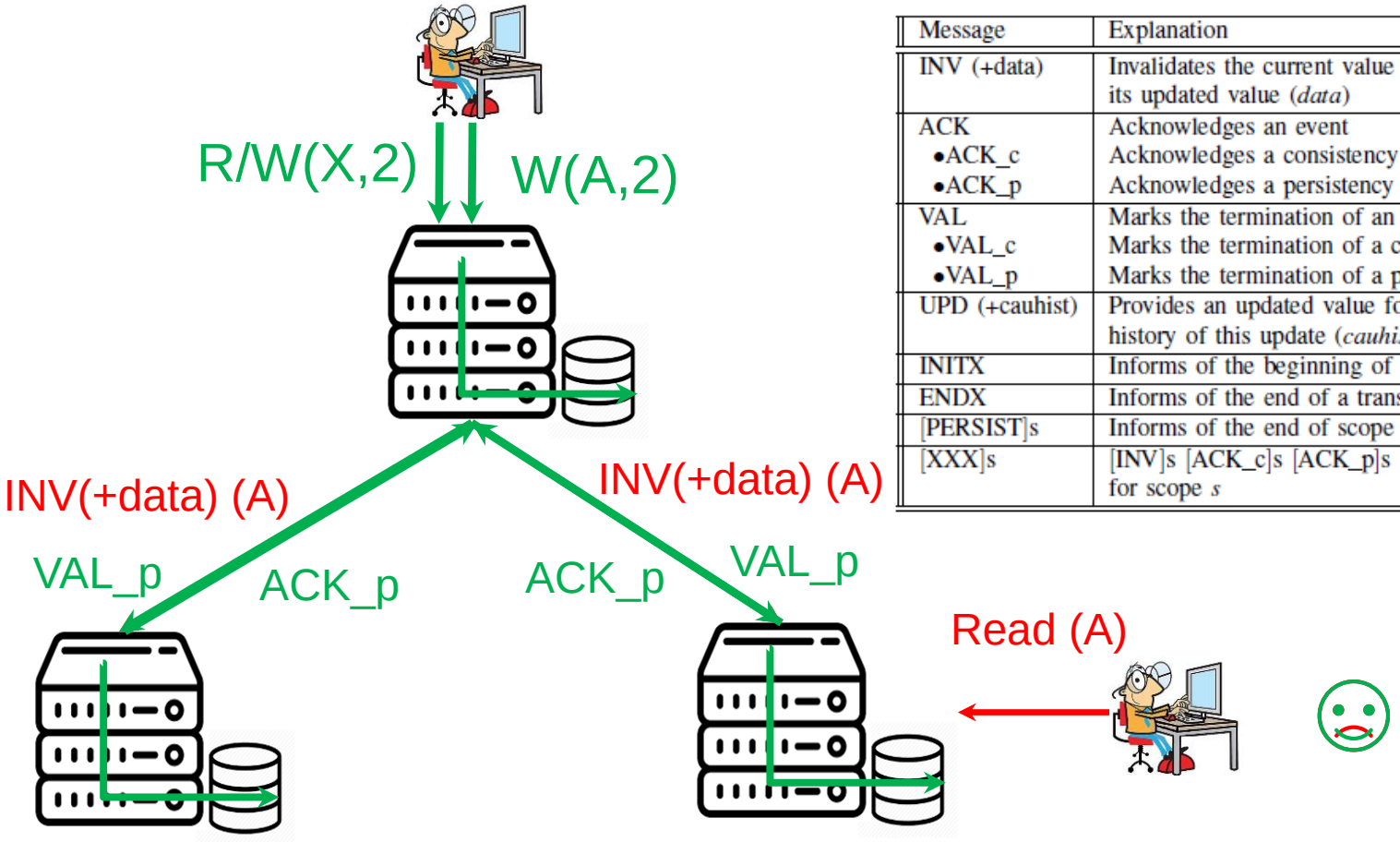
International Symposium on Microarchitecture (MICRO) 2021
October 2021

ILLINOIS

END

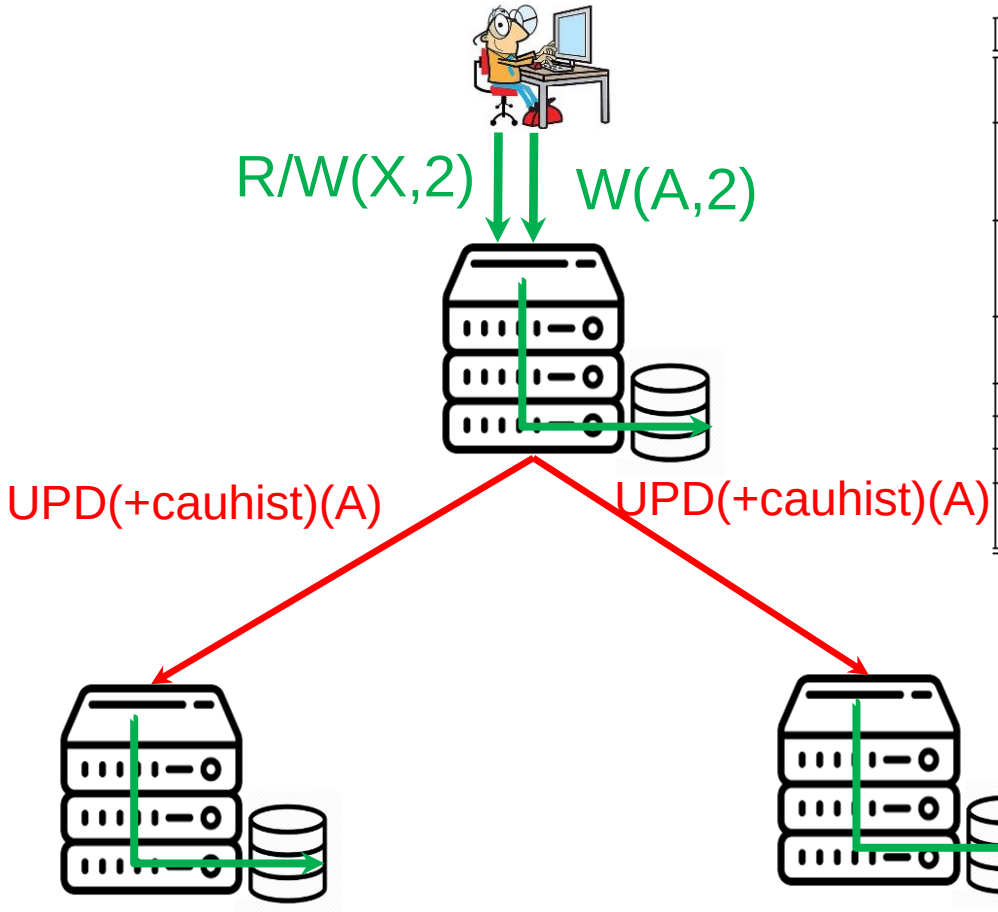
Backup Slides

DDP Protocols < Read-Enforced, Synchronous>



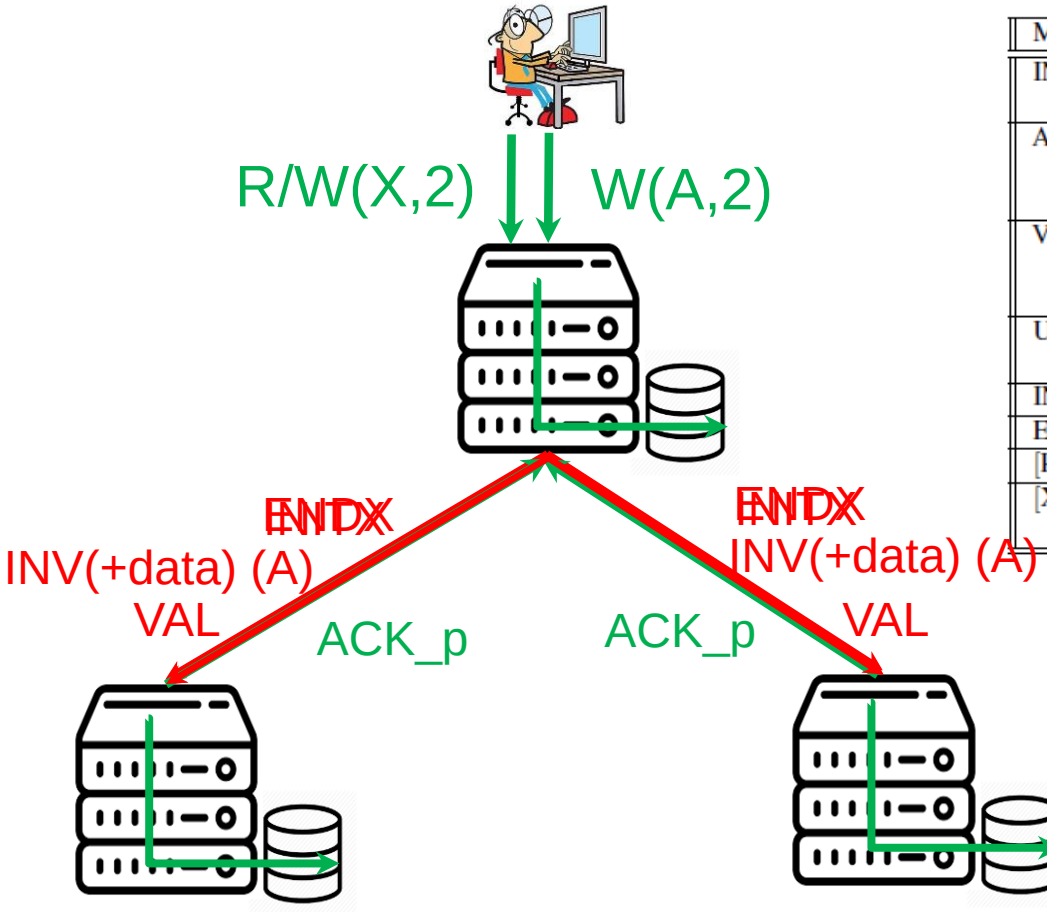
Message	Explanation
INV (+data)	Invalidates the current value of a key and provides its updated value (*data*)
ACK •ACK_c •ACK_p	Acknowledges an event Acknowledges a consistency event Acknowledges a persistency event
VAL •VAL_c •VAL_p	Marks the termination of an event Marks the termination of a consistency event Marks the termination of a persistency event
UPD (+cauhist)	Provides an updated value for a key plus the causal history of this update (*cauhist*)
INITX	Informs of the beginning of a transaction
ENDX	Informs of the end of a transaction
[PERSIST]s	Informs of the end of scope *s*
[XXX]s	[INV]s [ACK_c]s [ACK_p]s [VAL_c]s [VAL_p]s for scope *s*

DDP Protocols < Causal, Synchronous>



Message	Explanation
INV (+data)	Invalidates the current value of a key and provides its updated value (*data*)
ACK •ACK_c •ACK_p	Acknowledges an event Acknowledges a consistency event Acknowledges a persistency event
VAL •VAL_c •VAL_p	Marks the termination of an event Marks the termination of a consistency event Marks the termination of a persistency event
UPD (+cauhist)	Provides an updated value for a key plus the causal history of this update (*cauhist*)
INITX	Informs of the beginning of a transaction
ENDX	Informs of the end of a transaction
[PERSIST]s	Informs of the end of scope *s*
[XXX]s	[INV]s [ACK_c]s [ACK_p]s [VAL_c]s [VAL_p]s for scope *s*

DDP Protocols <TXN, Synchronous>



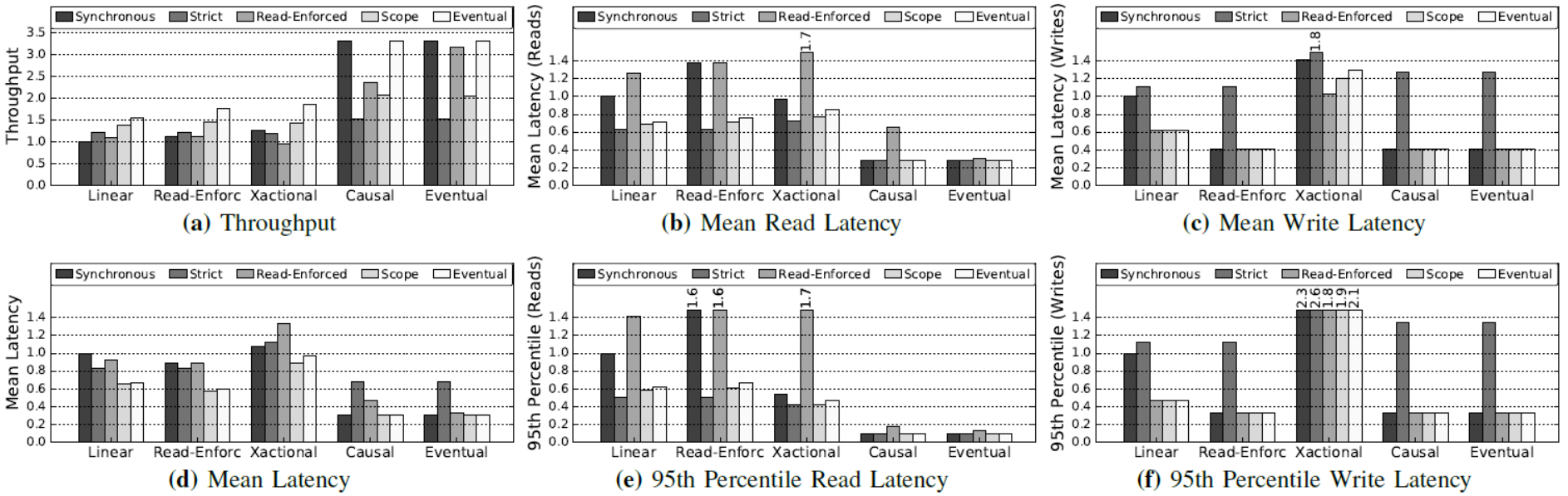
Message	Explanation
INV (+data)	Invalidates the current value of a key and provides its updated value (*data*)
ACK •ACK_c •ACK_p	Acknowledges an event Acknowledges a consistency event Acknowledges a persistency event
VAL •VAL_c •VAL_p	Marks the termination of an event Marks the termination of a consistency event Marks the termination of a persistency event
UPD (+cauhist)	Provides an updated value for a key plus the causal history of this update (*cauhist*)
INITX	Informs of the beginning of a transaction
ENDX	Informs of the end of a transaction
[PERSIST]s	Informs of the end of scope *s*
[XXX]s	[INV]s [ACK_c]s [ACK_p]s [VAL_c]s [VAL_p]s for scope *s*

DDP Tradeoffs

Cons.	Pers.	Dura-bility	Performance				Programmer Intuition			Progr.	Impl.
			Wr Opt?	Rd Opt?	Traffic	Overall	Monotonic Rds?	Non-Stale Rds?	Overall		
Linear	Synch.	↑	✗	✗	⟷	↓	✓	✓	↑	↑	↑
Read-Enf.	Synch.	⟷	✓	✗	⟷	⟷	✓	✗	⟷	↑	↑
Causal	Synch.	⟷	✓	✓	↑	↑	✓	✗	⟷	↑	↓
Event.	Sync.	↓	✓	✓	↓	↑	✗	✗	↓	↑	↑

- Weak consistency models sacrifice durability for performance
- Intuition is reduced → updates can be lost
- Causal is not easy to implement
  - Need to keep track of causality of updates
  - Communicate causal history to replicas

Evaluation - Performance

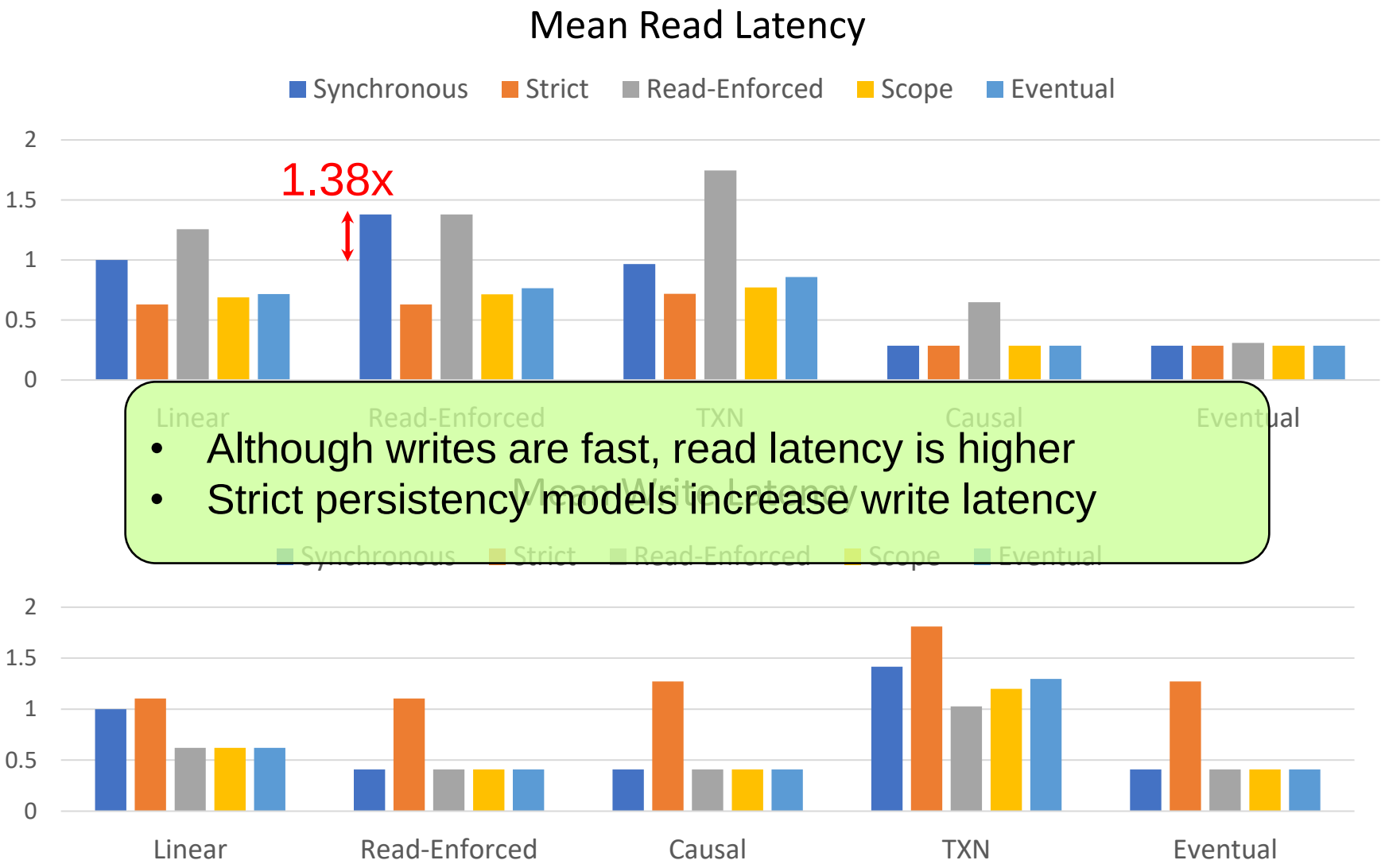


(a) Throughput (b) Mean Read Latency (c) Mean Write Latency

(d) Mean Latency (e) 95th Percentile Read Latency (f) 95th Percentile Write Latency

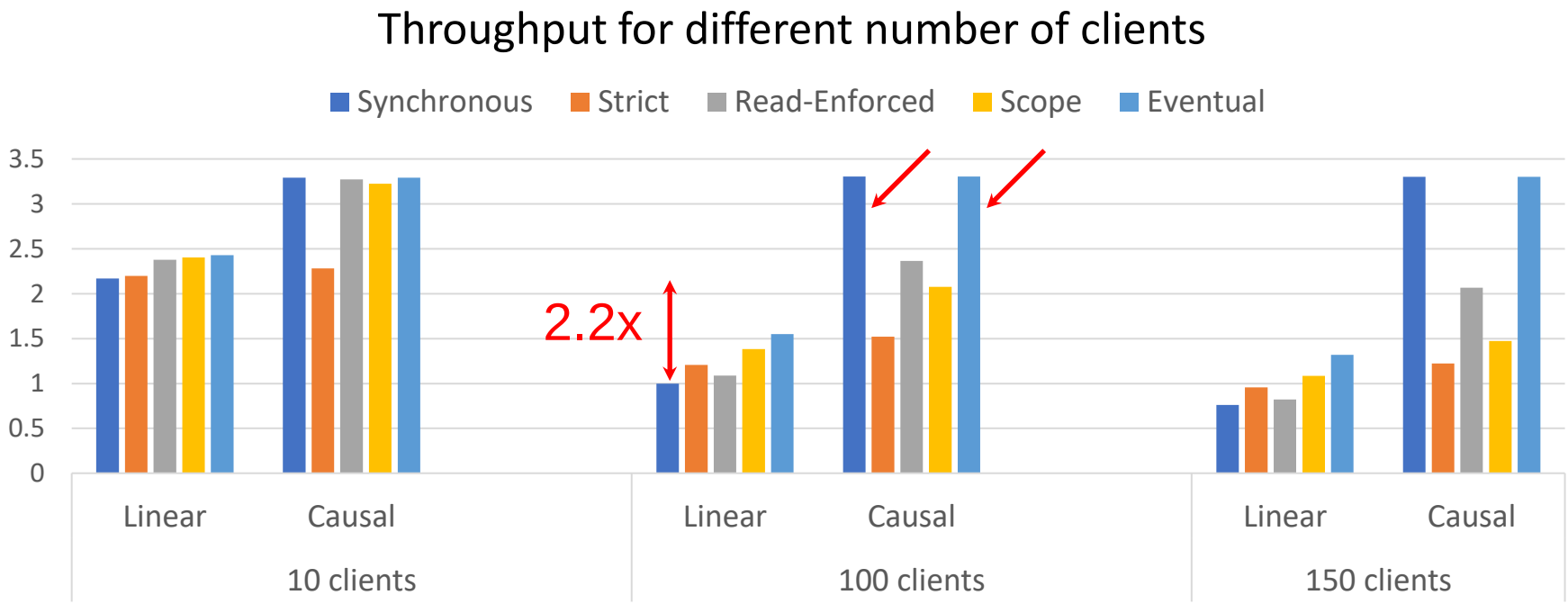
- Throughput is inversely correlated with read latency
- Read-Enforced Cons. in some cases suffers from high read latency and TXN from high conflicts
- Read-Enforced persistency is always better than Synchronous

Evaluation – Request Latency



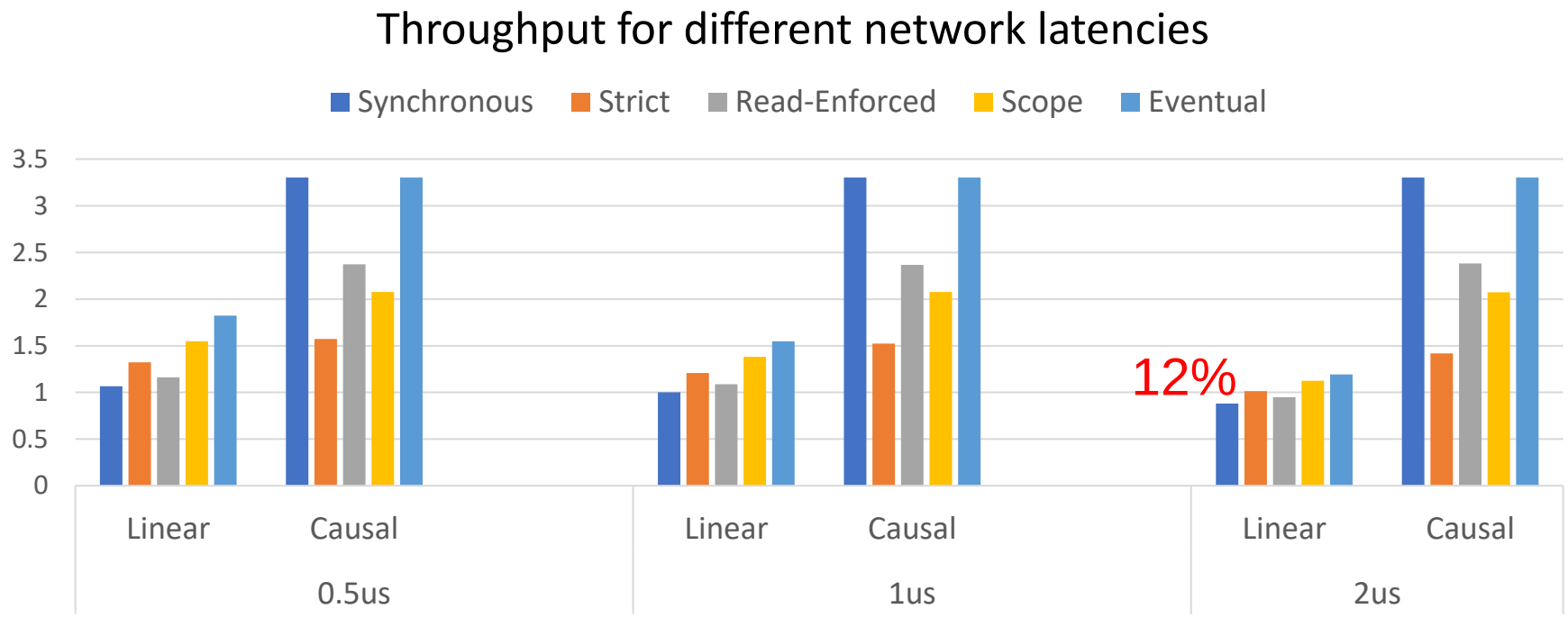
- Although writes are fast, read latency is higher
- Strict persistency models increase write latency

Evaluation – Number of Clients



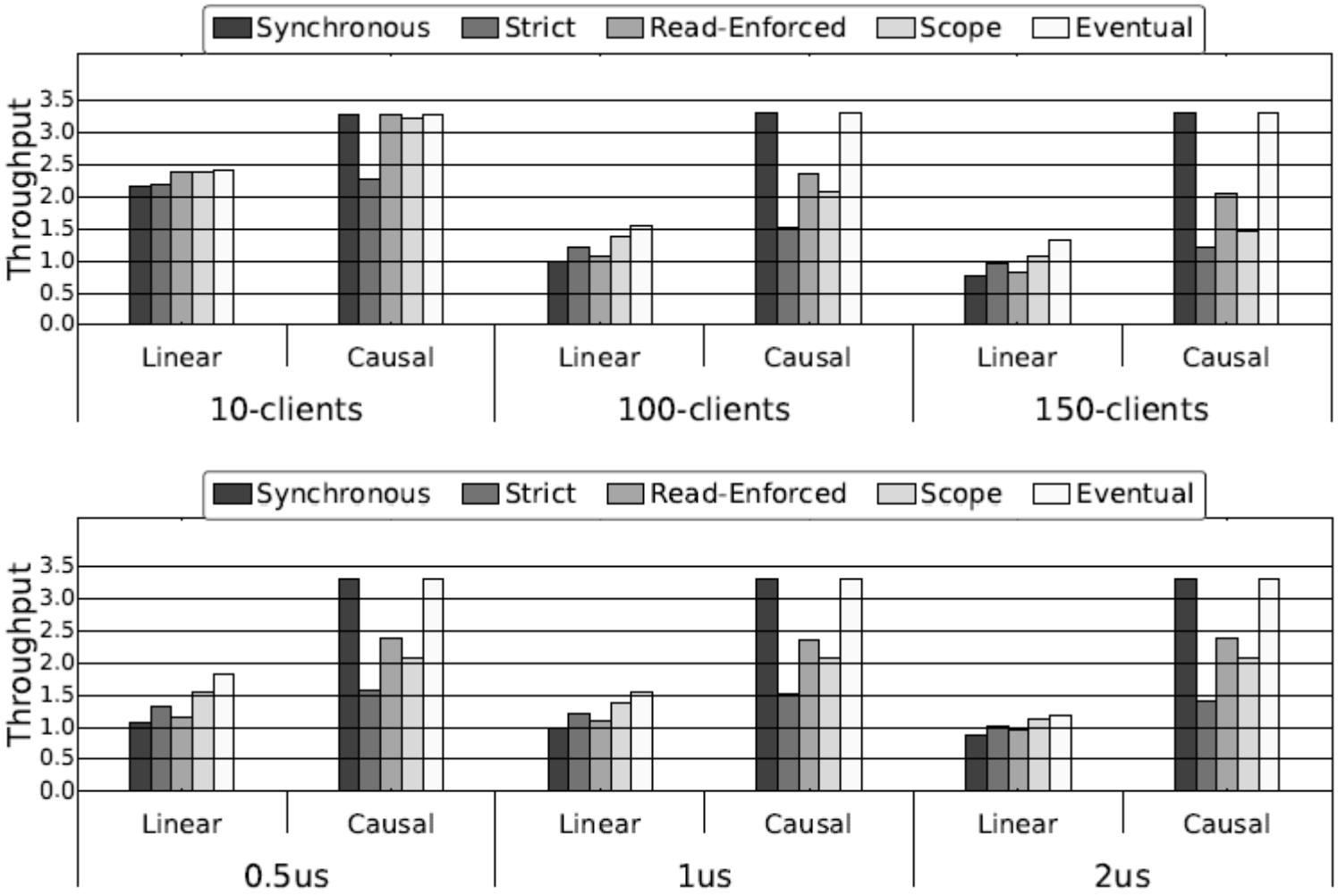
- Reducing the number of clients increases throughput
  - Decrease in conflicts and read stalls
- Causal is unaffected for Synchronous and Eventual pers
  - Reads do not stall

Evaluation – Network Latency



- Network latency affects mostly strict consistency models
  - Network is on the critical path of updates

Evaluation - Sensitivity



Evaluation - Sensitivity

- Workload-B: 95% reads & 5% writes
- Workload-A: 50% reads & 50% writes
- Workload-W: 5% reads & 95% writes