G-DUR
A Middleware for Assembling, Analyzing, and Improving Transactional Protocols

*Masoud Saeida Ardekani**, INRIA & UPMC-Lip6
Pierre Sutra, Université de Neuchâtel
Marc Shapiro, INRIA & UPMC-Lip6

* now @ Purdue University

Motivation

- Understanding transactional protocols
- Perform apples-to-apples comparison
- Study their bottlenecks
- Improve them

Insight

- Read an object
  - Read the latest committed version
  - Take a consistent snapshot
  - Take the latest consistent snapshot

- Certification
  - No write-write and read-write conflict
  - No write-write conflict
- Atomic Commitment
  - 2PC
  - Atomic Broadcast
  - Atomic Multicast + Voting
- Commutativity
  - commute if no read-write nor write-write conflict
  - commute if no write-write conflict

Insight

- Read an object
  - Read the latest committed version
  - Take a consistent snapshot
  - Take the latest consistent snapshot

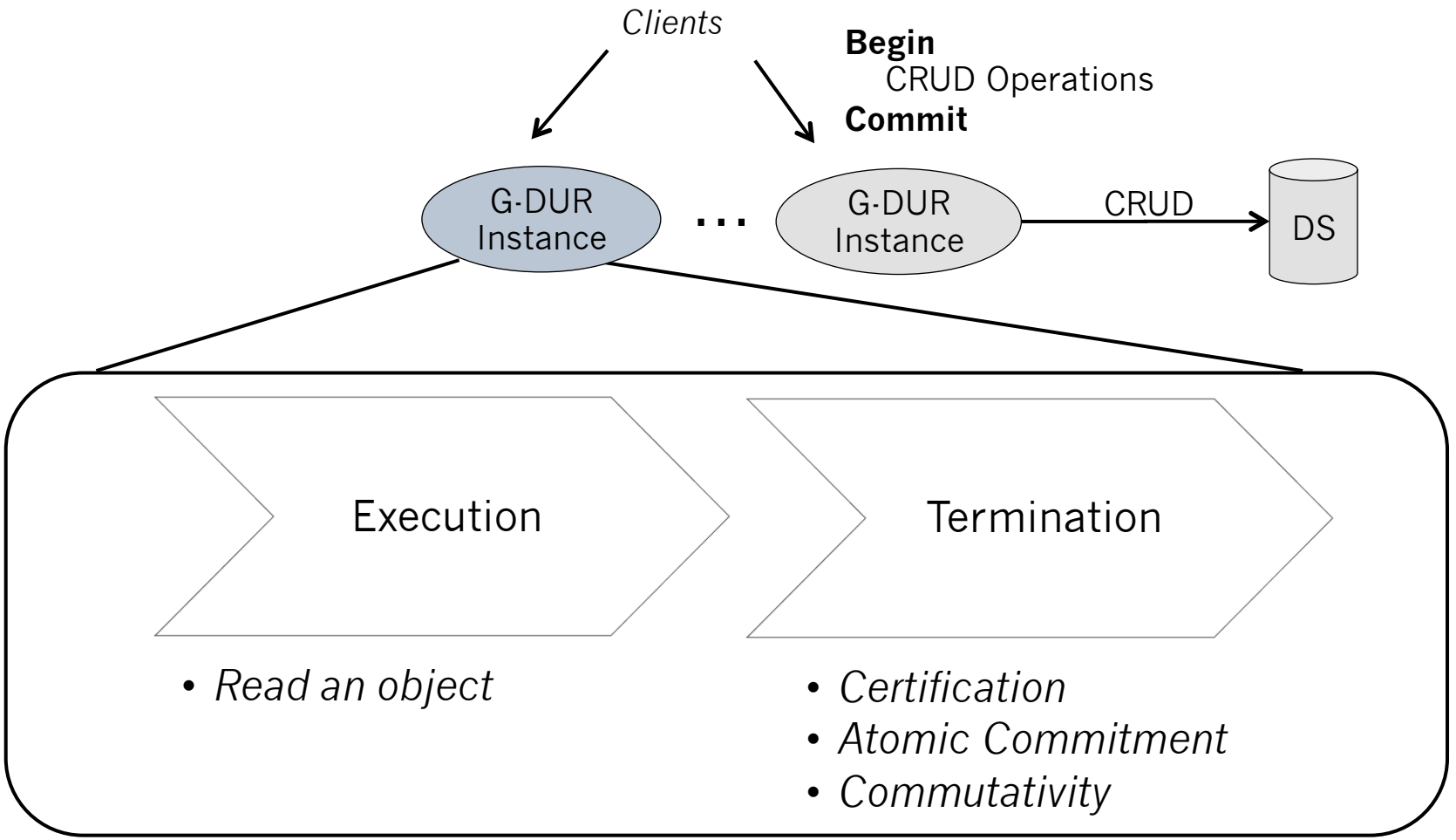
- Certification
  - No write-write and read-write conflict
  - No write-write conflict
- Atomic Commitment
  - 2PC
  - Atomic Broadcast
  - Atomic Multicast + Voting
- Commutativity
  - commute if no read-write nor write-write conflict
  - commute if no write-write conflict

Insight

- Read an object
  - Read the latest committed version
  - Take a consistent snapshot
  - Take the latest consistent snapshot

- Certification
  - No write-write and read-write conflict
  - No write-write conflict
- Atomic Commitment
  - 2PC
  - Atomic Broadcast
  - Atomic Multicast + Voting
- Commutativity
  - commute if no read-write nor write-write conflict
  - commute if no write-write conflict

G-DUR:
Generic Deferred Update Replication



Programming Effort

Realization Point	P-Store [SRDS'10]	GMU [ICDCS'12]
Read Object	Read the latest version of objects	Read the latest consistent snapshot
Atomic Com.	Atomic Multicast + Voting	2PC

Certify	No read-write or write-write conflict	If (write-set is empty)
		Commit
		else
		No read-write or write-write confclit

Commutativity	No read-write or write-write conflict	No read-write or write-write conflict

Ease of Programming

Protocol		Source Lines of Code	
	Consistency	**G-DUR**	**Original**
P-Store [SRDS'10]	Serializability	179	6000 [Java]
S-DUR [DSN'12]	Serializability	397	N/A
GMU [ICDCS'12]	Update Serializability	476	6000 [Java]
Serrano [PRDC'07]	Snapshot Isolation	351	N/A
Walter [SOSP'11]	Parallel Snapshot Isolation	599	30000 [C++]
Jessy [SRDS'13]	Non-monotonic Snapshot Isolation	352	6000 [Java]

Framework SLOC ~ 20'000

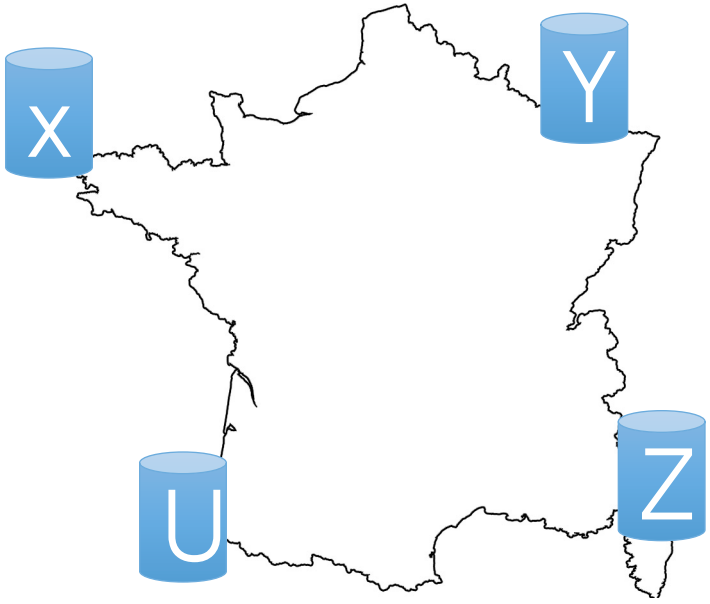
Case Study

- Apples-to-apples Comparison
- Study bottlenecks and limitations of protocol
- Improving Protocol
- Compare degrees of dependability

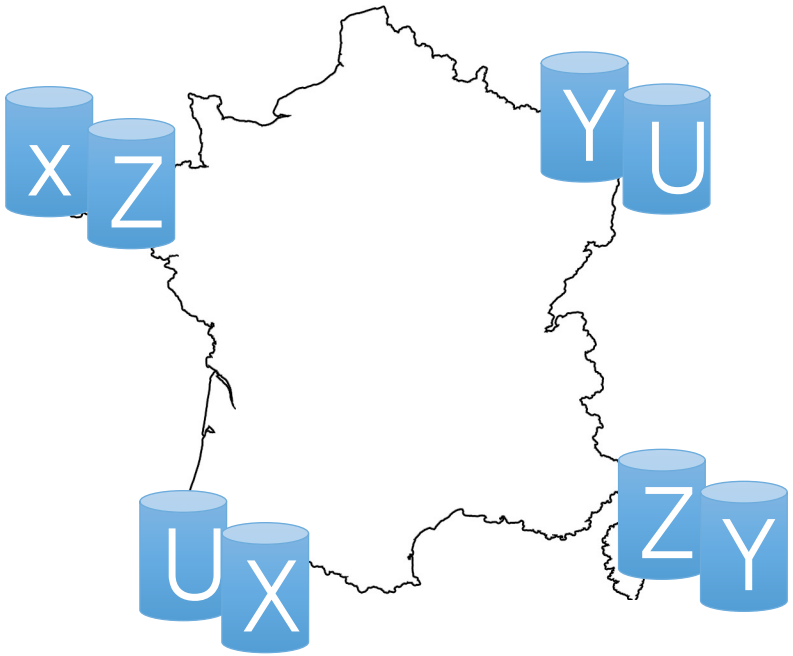
Evaluation Setup

- 4 Sites in Grid'5000
- Clients distributed uniformly among sites
- Modified YCSB benchmark [SOCC'10]

Disaster Prone

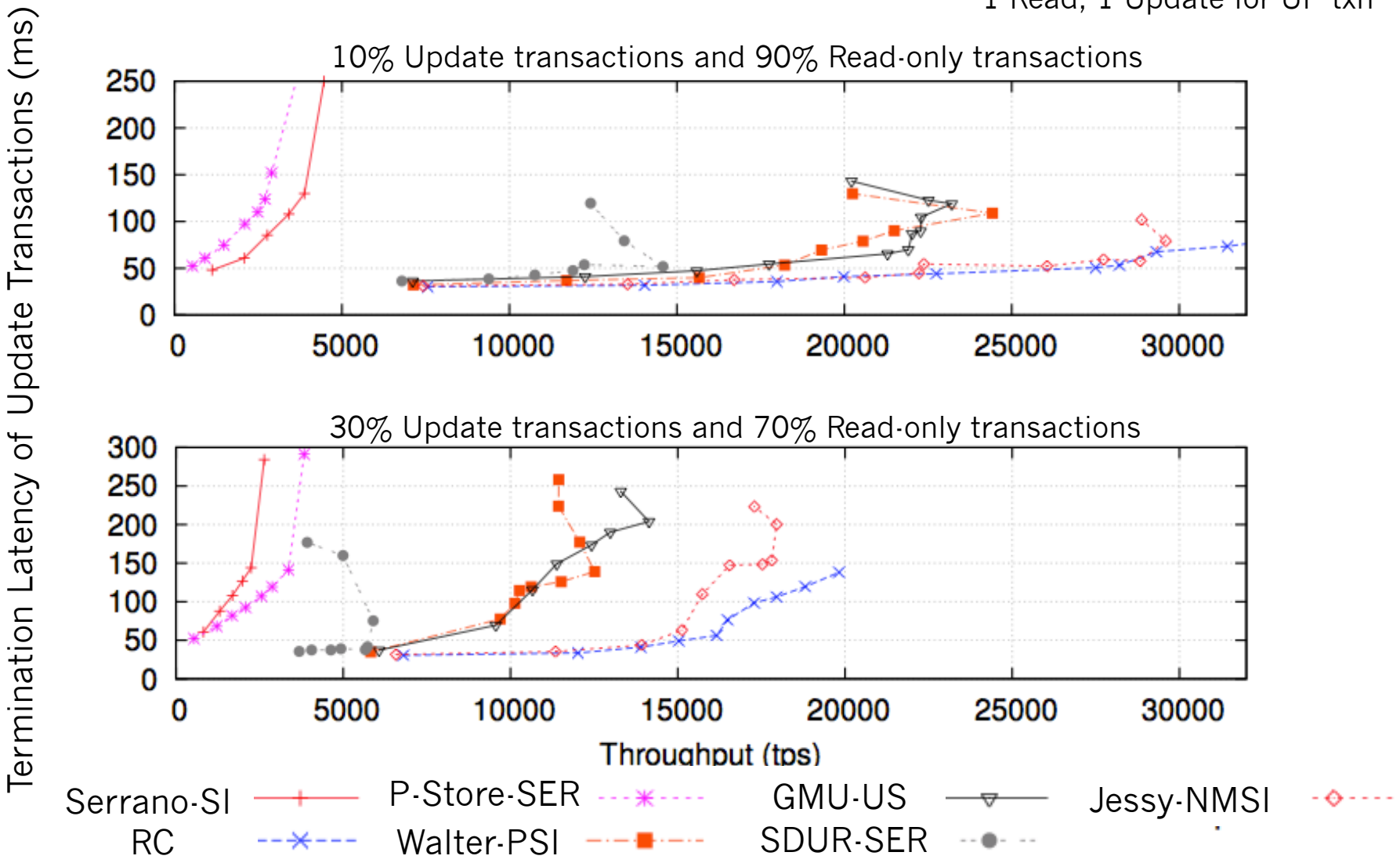


Disaster Tolerant

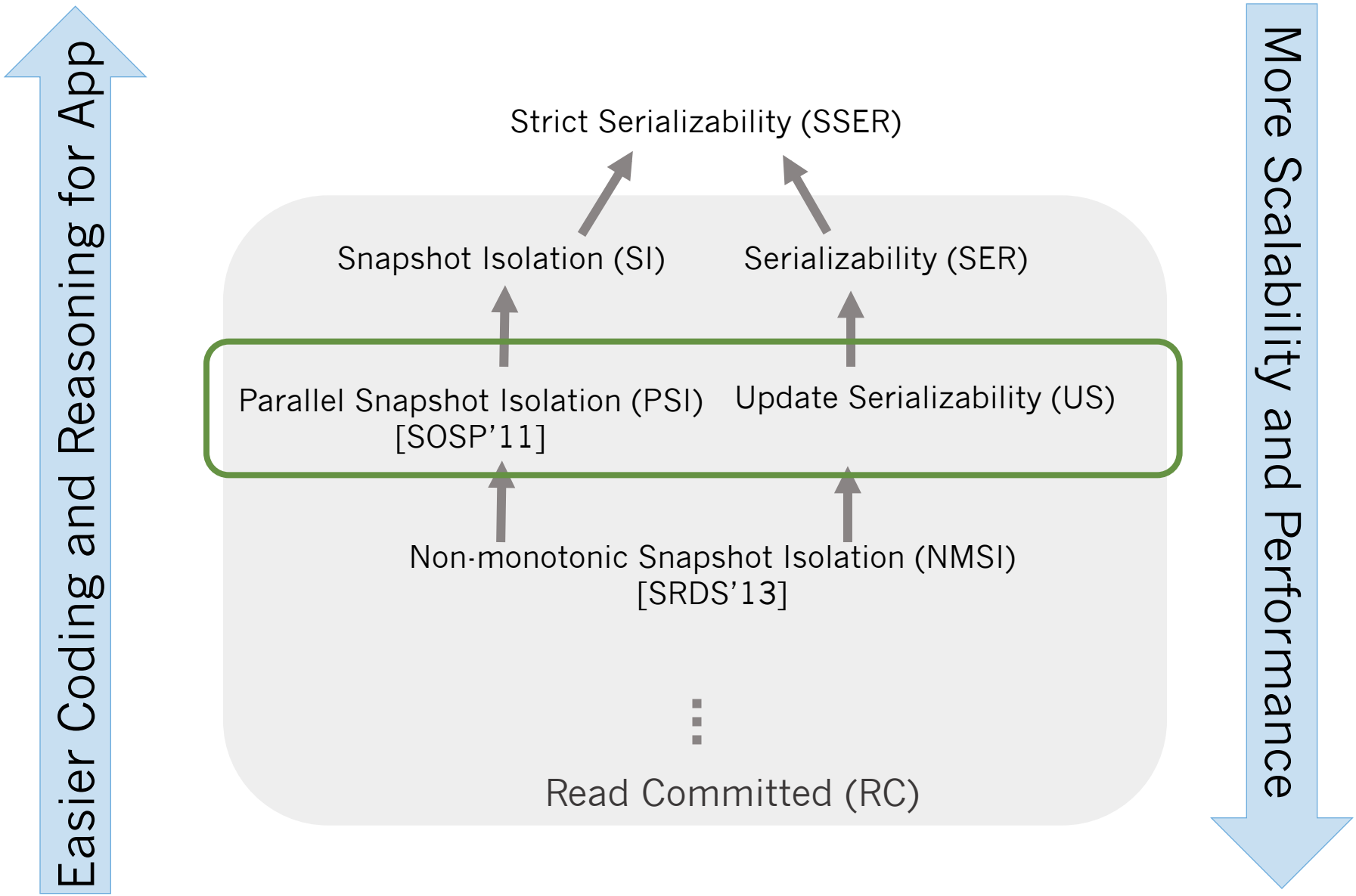


Comparing Protocols in Disaster Prone Rep.

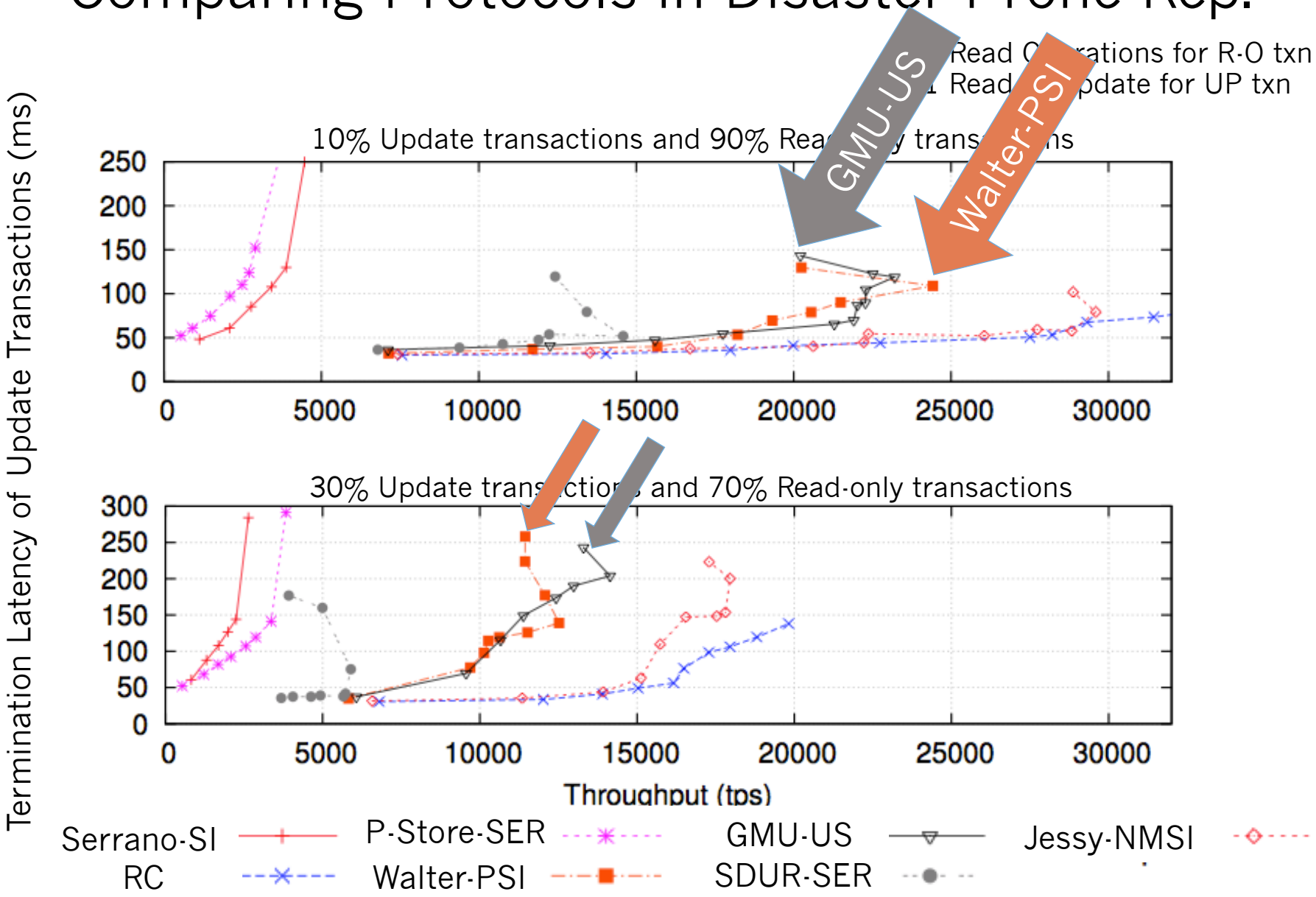
2 Read Operations for R-O txn
1 Read, 1 Update for UP txn



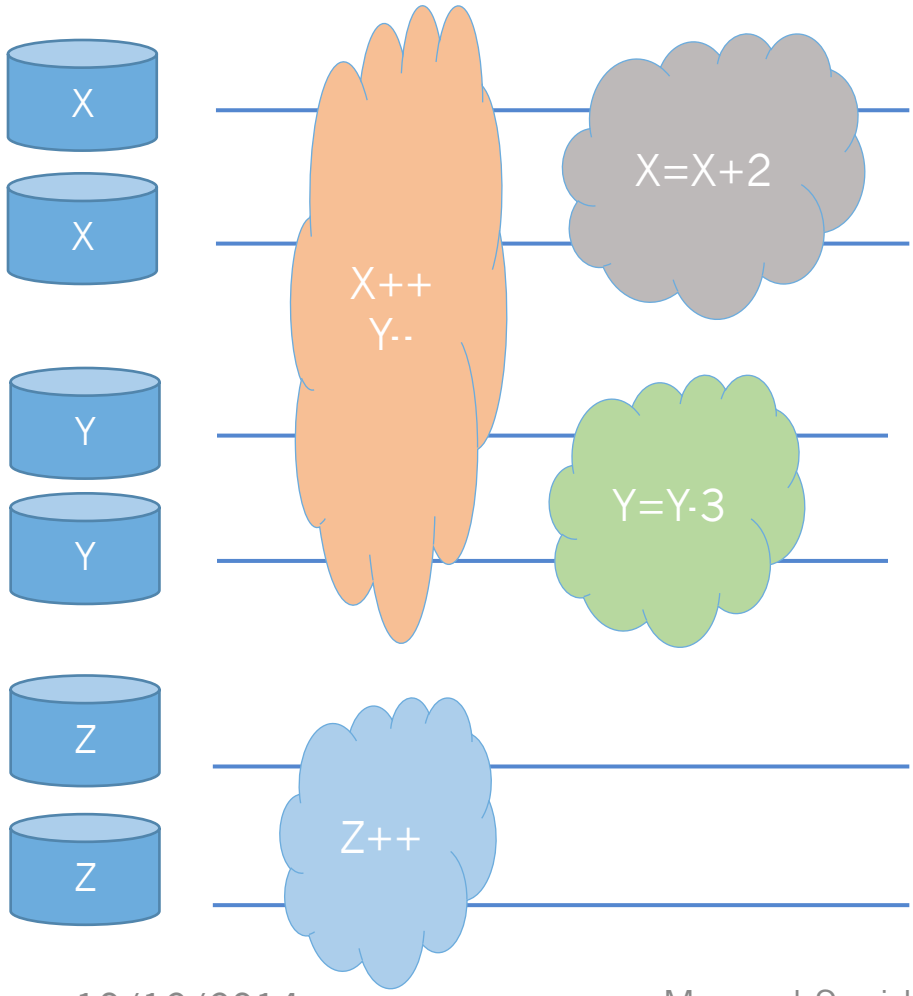
Consistency (Isolation level) Hierarchy



Comparing Protocols in Disaster Prone Rep.

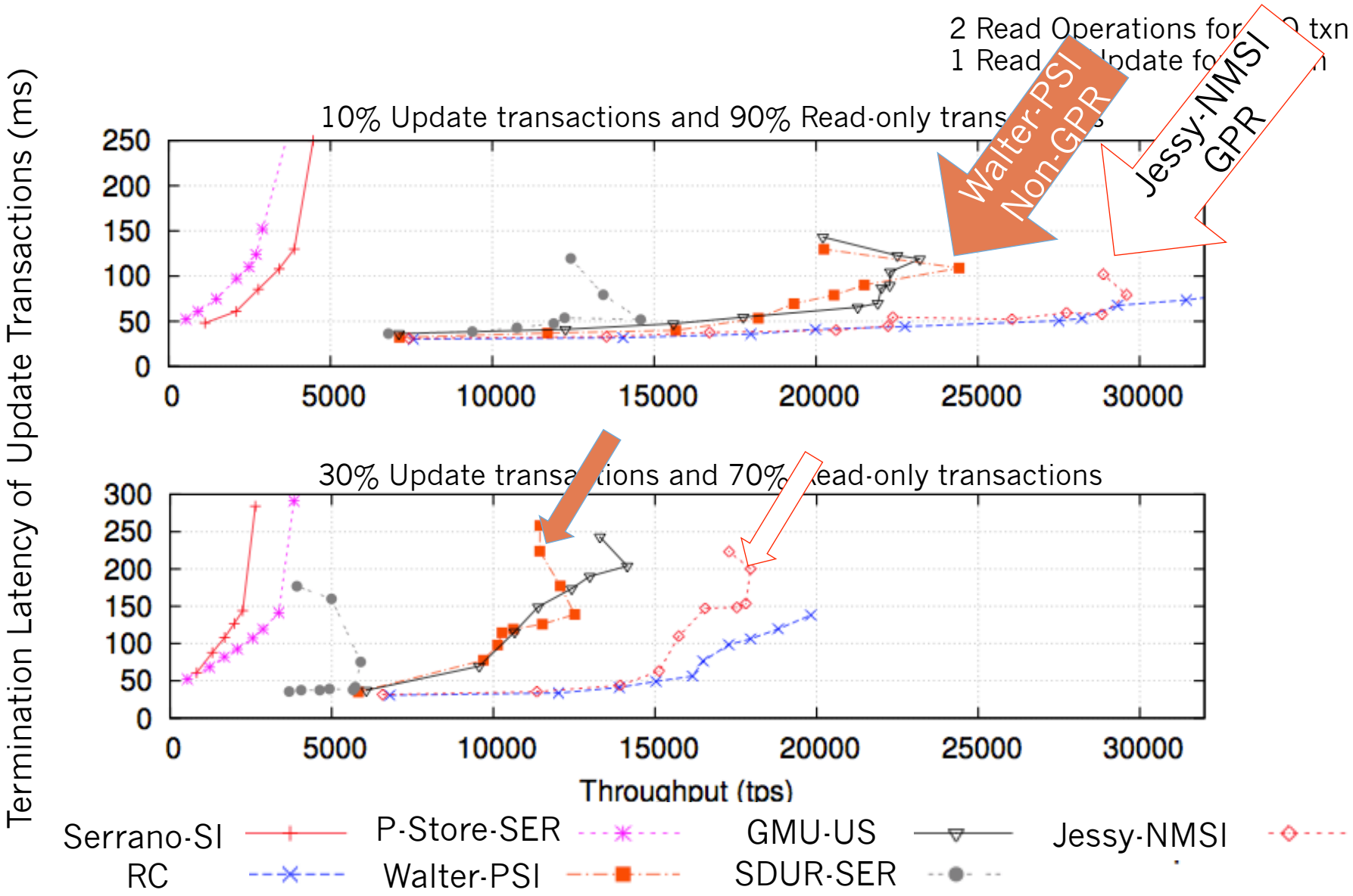


Genuine Partial Replication (GPR) [Schiper'10]

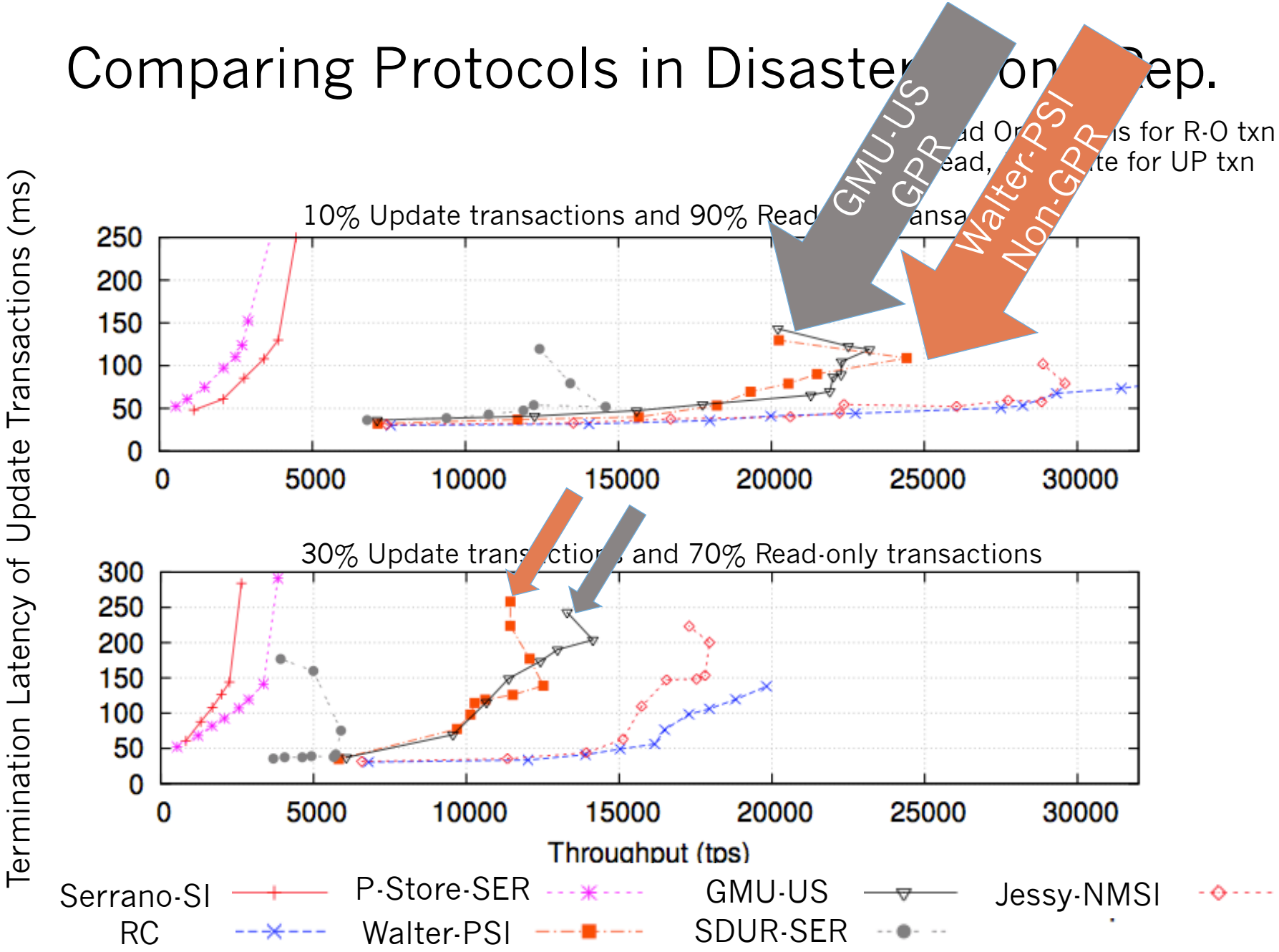


- Only replicas of objects read or written inside transaction communicate
- Non-conflicting transactions do not interfere ➔ More Parallelism
- Disjoint-Access-Parallelism
- In the next slide we compare a GPR protocol with a non-GPR protocol

Comparing Protocols in Disaster Prone Rep.



Comparing Protocols in Disaster-Prone Rep.

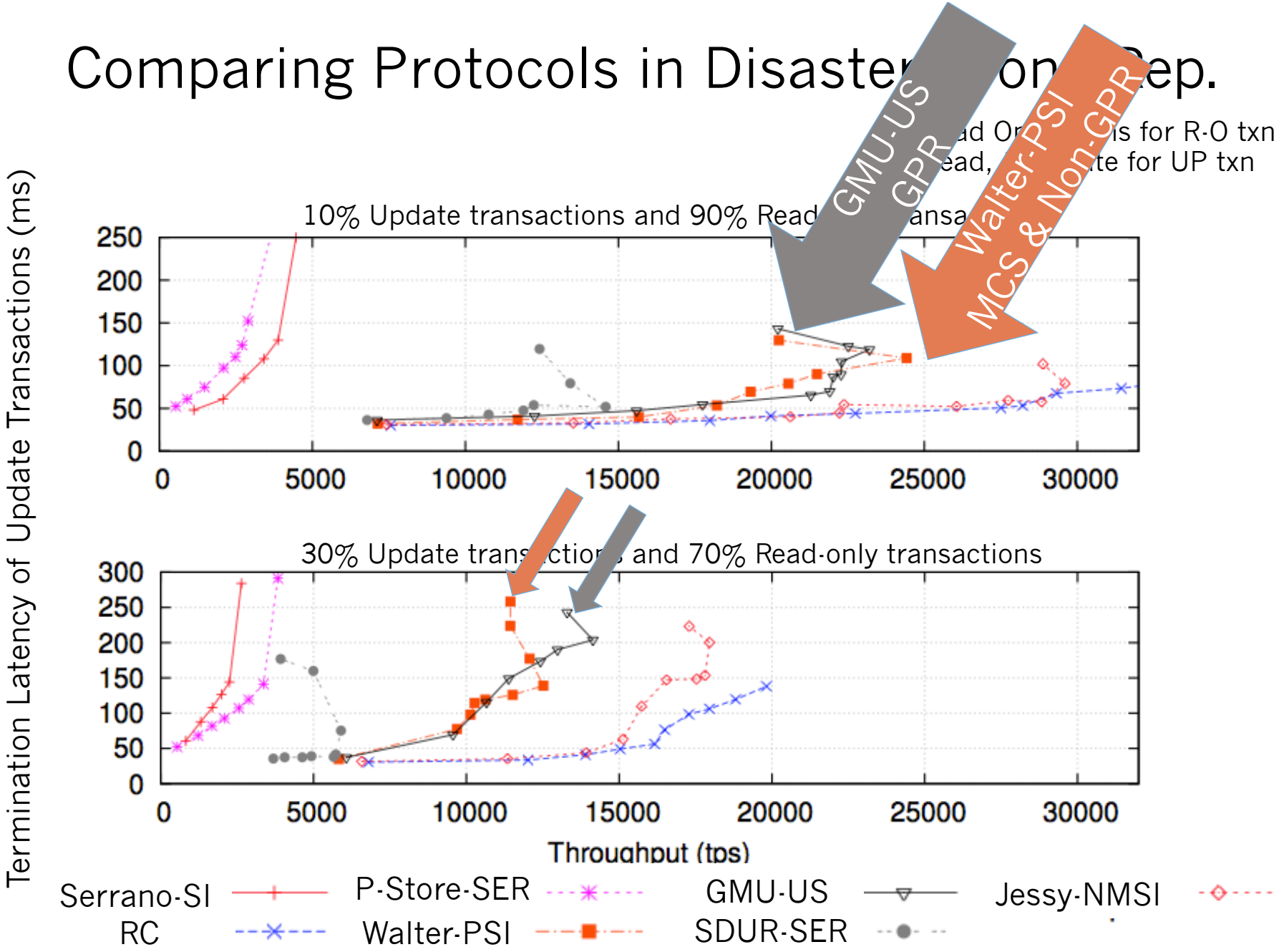


Minimal Commitment Sync. (MCS)

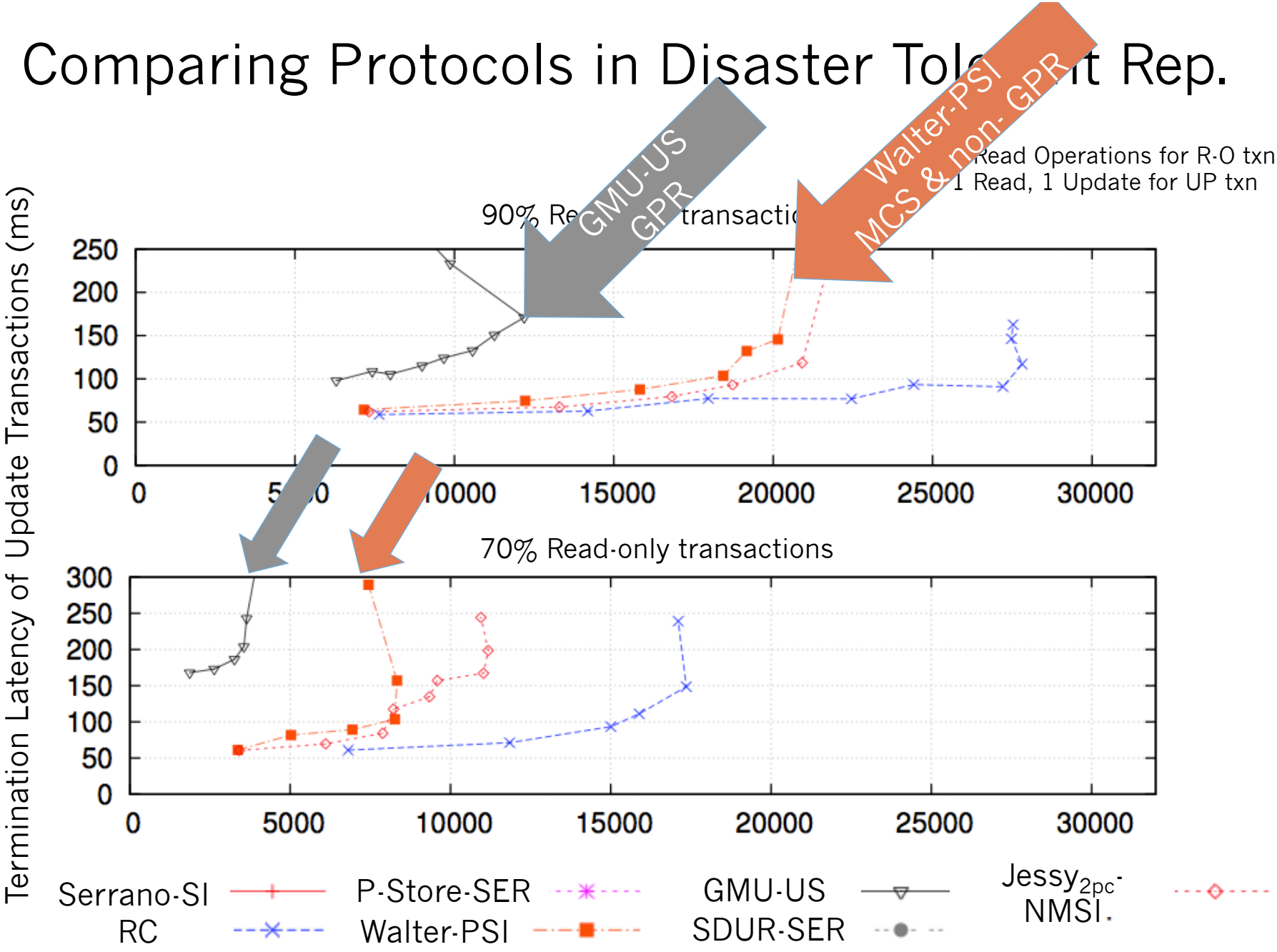
- Abort a transaction in case of
  - a read-write or write-write conflict
  - write-write conflict

- In the next slide we study the effect of MCS

Comparing Protocols in Disaster



Comparing Protocols in Disaster Tole[illegible]t Rep.



Summary / Conclusion

- Many transactional protocols follow DUR approach

- G-DUR allows fast prototyping of protocols to:
  - Compare protocols
  - study limitations & bottlenecks of a protocol
  - In the paper, compare degrees of dependability

- Common specification language