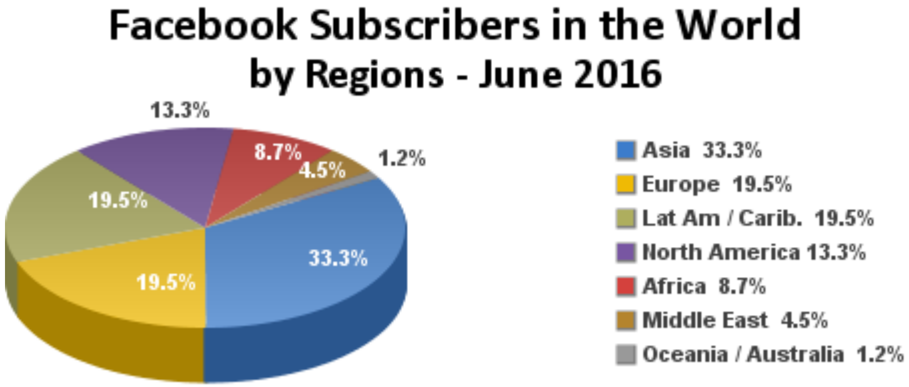
Fine-grained consistency for geo-replicated systems

Cheng Li, Nuno Preguica, Rodrigo Rodrigues

University of Science and Technology of China
NOVA LINCS & FCT, Univ. NOVA de Lisboa
INESC-ID & Instituto Superior Técnico, Universidade de Lisboa

Unprecedented growth in Internet services

- As of June 2017 , Facebook has 2 billion monthly active users.



Source: Internet World Stats - www.internetworldstats.com/facebook.htm
Basis: 1,679,433,530 Internet users on June 30, 2016

Geo-users demand instant responses

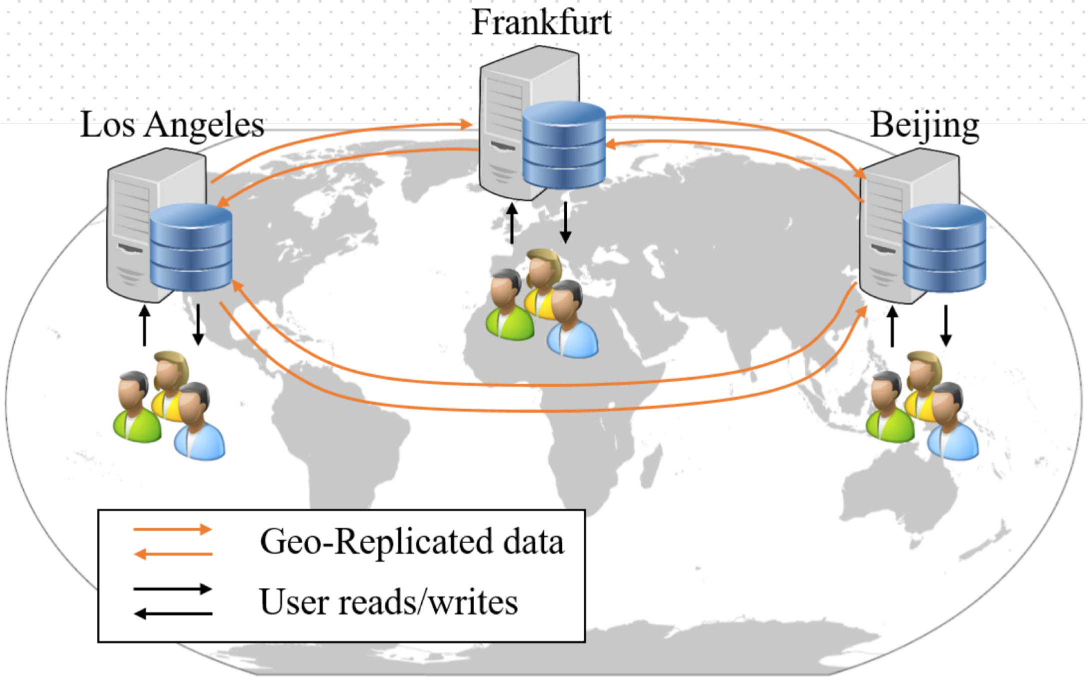
	Distinct Queries/User	Query Refinement	Revenue/User	Any Clicks	Statisfaction	Time to Click (increase in ms)
50ms	-	-	-	-	-	
200ms	-	-	-	-0.3%	-0.4%	500
500ms	-	-0.6%	-1.2%	-1.0%	-0.9%	1200
1000ms	-0.7%	-0.9%	-2.8%	-1.9%	-1.6%	1900
2000ms	-1.8%	-2.1%	-4.3%	-4.4%	-3.8%	3100

- Strong negative impact of delay on user activities [1]
- Google counts site speed as a ranking factor [2].

[1] E. Schurman and J. Brutlag, "Performance Related Changes and their User Impact". Talk at Velocity '09
[2] https://searchengineland.com/google-now-counts-site-speed-as-ranking-factor-39708

Geo-Replication helps

- *Performance*: local reads
- *Availability*: data still available unless all replicas fail or become unreachable
- *Scalability*: load balance across sites for reads



Fundamental trade-offs

USTC, CHINA ADSLAB

Strong consistency (SC)

e.g., Paxos [TOCS'98]

- ✓ *State convergence*
- ✓ *Invariant preservation*
- ✗ *High latency*
- ✗ *Low throughput*

Eventual consistency (EC)

e.g., Dynamo [SOSP'07]

- ✓ *Low latency*
- ✓ *High throughput*
- ✗ *State divergence*
- ✗ *Invariant violation*

Our prior work

RedBlue Consistency [OSDI'12, ATC'14]
allows operations to be executed under either strong or eventual consistency.

Strong consistency (SC)

e.g., Paxos [TOCS'98]

- ✓ *State convergence*
- ✓ *Invariant preservation*

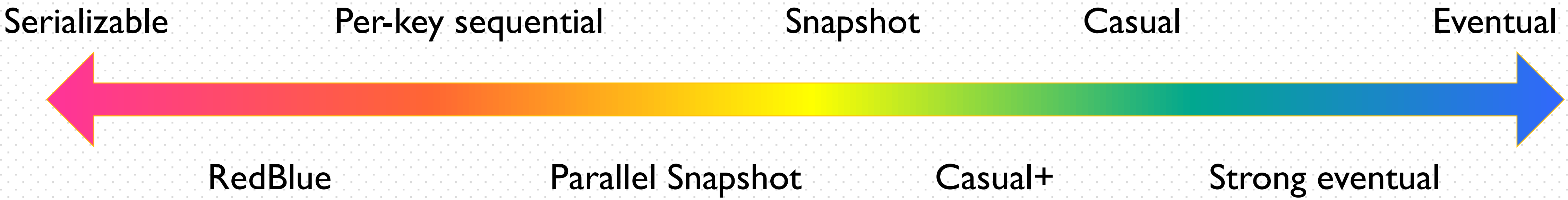
Eventual consistency (EC)

e.g., Dynamo [SOSP'07]

- ✓ *Low latency*
- ✓ *High throughput*

Coarse-grained classification may add unnecessary coordination!

Consistency spectrum



- Too many consistency models, some of which have subtle differences
- Need a unified consistency framework to capture all these semantics

Outline

1. Background and problem statement
2. Partial-Order Restrictions (PoR) Consistency
3. Olisipo: PoR consistent coordination service
4. Evaluation and results
5. Conclusion

Geo-replicated auction service

winner	bidder	price
	Bob	10

winner	bidder	price

US

UK

bid($10)

Geo-replicated auction service

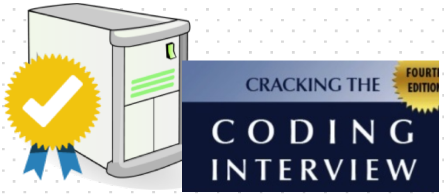
winner

bidder	price
Bob	10

winner

bidder	price
Alice	15

US



UK

bid($15)

Geo-replicated auction service

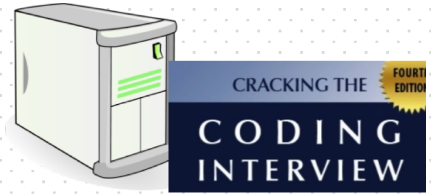
winner
Bob

bidder	price
Bob	10

winner

bidder	price
Alice	15

US



UK

close()

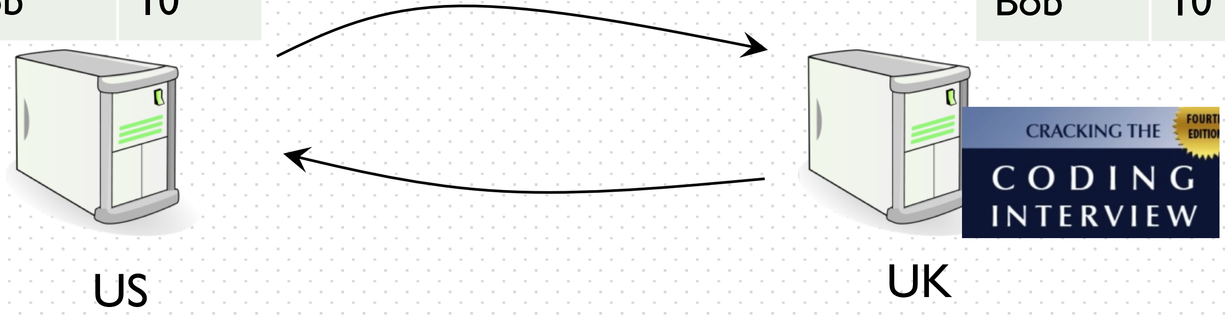
Geo-replicated auction service

winner
Bob

bidder	price
Alice	15
Bob	10

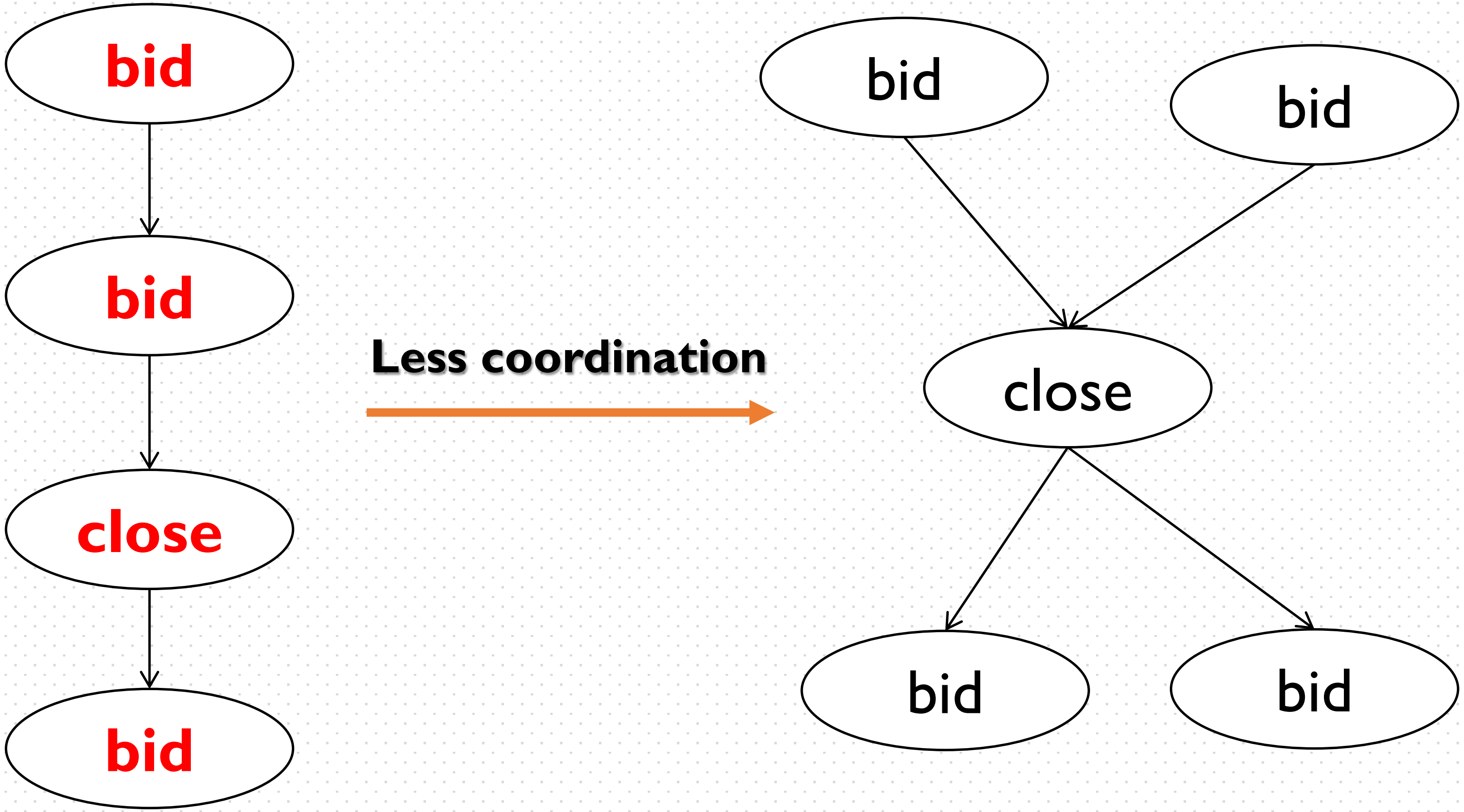
winner
Bob

bidder	price
Alice	15
Bob	10



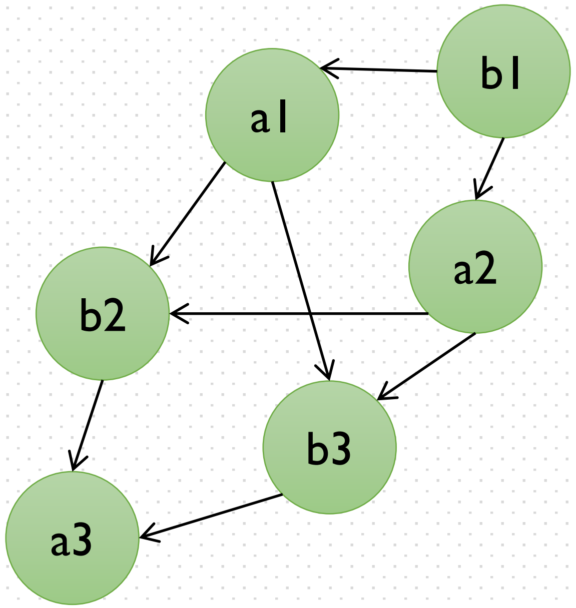
Bob won even with a lower bid than Alice.

Fine-grained coordination



Visibility restrictions

- A restriction between two operations implies that one must see effects introduced by the other.
- For operation a, b, the restriction $r(a, b)$ implies that $a \prec b \lor b \prec a$ w.r.t any partial order $\prec$.



If $a \prec b \lor b \prec a$,
then $r(a, b)$ is met in $\prec$.

Partial order-restrictions (PoR) Consistency

USTC, CHINA ADSLAB

- A geo-replicated system S is associated with a set of restrictions Rs.
- S is ***PoR Consistent*** if, for any its executions, there exists an admissible partial order, where all restrictions in Rs are met.

Partial order-restrictions (PoR) Consistency

- A geo-replicated system S is associated with a set of restrictions Rs.
- S is ***PoR Consistent*** if, for any its executions, there exists an admissible partial order, where all restrictions in Rs are met.

Fewer restrictions → Weaker consistency

Tunable (parameterized) consistency model

Partial order-restrictions (PoR) Consistency

USTC, CHINA ADSLAB

- A geo-replicated system S is associated with a set of restrictions Rs.
- S is ***PoR Consistent*** if, for any its executions, there exists an admissible partial order, where all restrictions in Rs are met.

Fewer restrictions → Weaker consistency

Tunable (parameterized) consistency model

Causal consistency
Rs = {}

RedBlue consistency
Rs = {r(a,b) | a, b are red operations}

Serializability
Rs = {r(a,b) | for any pair of operations a, b}

Challenges of adopting PoR

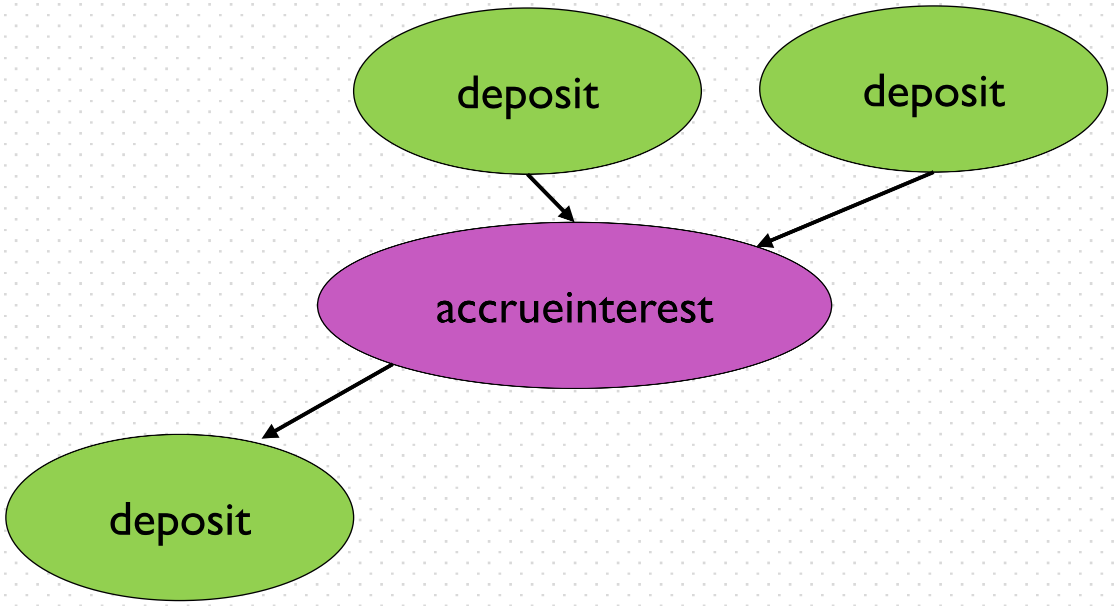
- What are the set of restrictions to be added?
  - They must ensure relevant properties, e.g., state convergence, invariant preservation.
- Is the set of added restrictions minimal?
  - i.e., no unnecessary coordination

State convergence

- If all replicas execute the same set of operations then they reach the same state
- Must place a restriction over any pair of non-commuting operations
- Consider a geo-replicated bank example

```
deposit(float m){
    balance = balance + m;
}

accrueinterest(){
    float delta=balance × interest;
    balance=balance + delta;
}
```



Invariant preservation

- Insight: for any violation, add restrictions among a *minimal* set of *concurrent* conflicting operations
  - i.e., removing any conflicting op, violation disappears
  - named as "I-conflict set"

Invariant preservation

Invariant: $\exists\, winner \rightarrow winner.bid \text{ is highest in } bidTable$

$\boldsymbol{bid(uId, bid)}$

$\boldsymbol{close(wId)}$

Invariant preservation

Invariant: $\exists\ winner \ \rightarrow winner.bid\ is\ highest\ in\ bidTable$

Weakest precondition

$\{auction\ is\ Open\}$

$\boldsymbol{bid(uId, bid)}$

$\{bidTable = bidTable \cup \{< uId, bid >\}\}$

$\{\exists\, w \ \in\ bidTable.w.uId\ =\ wId\ \wedge\ w.bid\ is\ highest\}$

$\boldsymbol{close(wId)}$

$\{winner = wId\ \wedge\ auction\ is\ Closed\}$

Postcondition

Invariant preservation

Invariant: $\exists\ winner \rightarrow winner.bid\ is\ highest\ in\ bidTable$

$\{auction\ is\ Open\}$	$\{\exists\ \boldsymbol{w} \in \boldsymbol{bidTable.w.uId} = \boldsymbol{wId} \wedge \boldsymbol{w.bid\ is\ highest}\}$
$\boldsymbol{bid}(\boldsymbol{uId}, \boldsymbol{bid})$	$\boldsymbol{close}(\boldsymbol{wId})$
$\{bidTable = bidTable \cup \{< uId, bid >\}\}$	$\{winner = wId \wedge auction\ is\ Closed\}$

Invalidation

- {*close*, *bid*} is an "I-conflict set".
- The restriction *r*{*close*, *bid*} must be enforced!

Outline

1. Background and problem statement
2. Partial-Order Restrictions (PoR) Consistency
3. Olisipo: PoR consistent coordination service
4. Evaluation and results
5. Conclusion

Olisipo - Design rationale

Give a restriction $r(a, b)$

- Workload 1: a and b have the same prevalence

- Workload 2: a occurs more often than b

Olisipo - Design rationale

Give a restriction $r(a, b)$

- Workload 1: a and b have the same prevalence

Symmetry protocol: Every a (b) instance acquires a permission from a centralized server w.r.t all concurrent b (a) instances.

- Workload 2: a occurs more often than b

Olisipo - Design rationale

Give a restriction $r(a, b)$

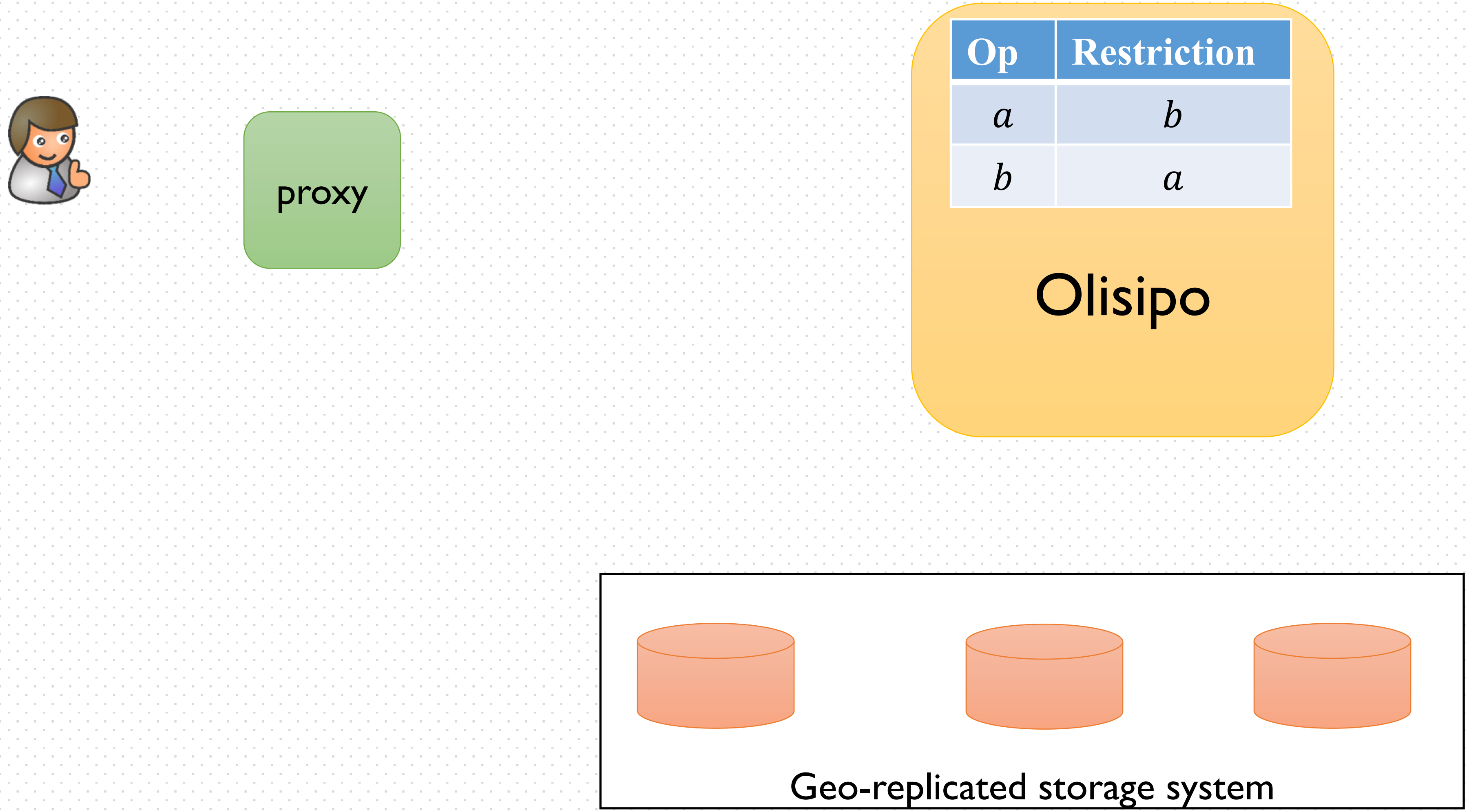
- Workload 1: a and b have the same prevalence

Symmetry protocol: Every a (b) instance acquires a permission from a centralized server w.r.t all concurrent b (a) instances.

- Workload 2: a occurs more often than b

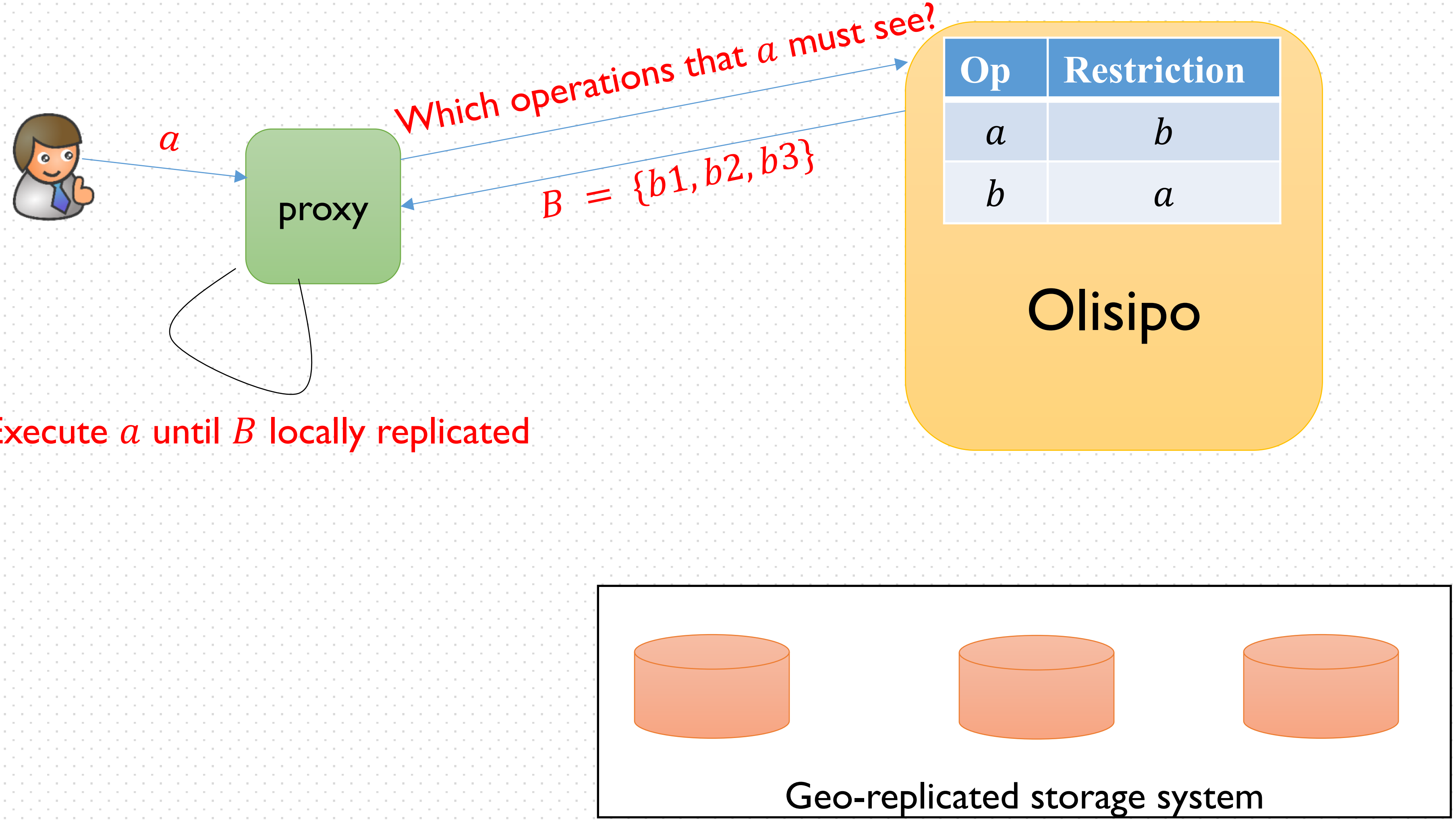
Asymmetry protocol: Every b instance acts as a global barrier w.r.t all concurrent a instances.

Olisipo - Overview



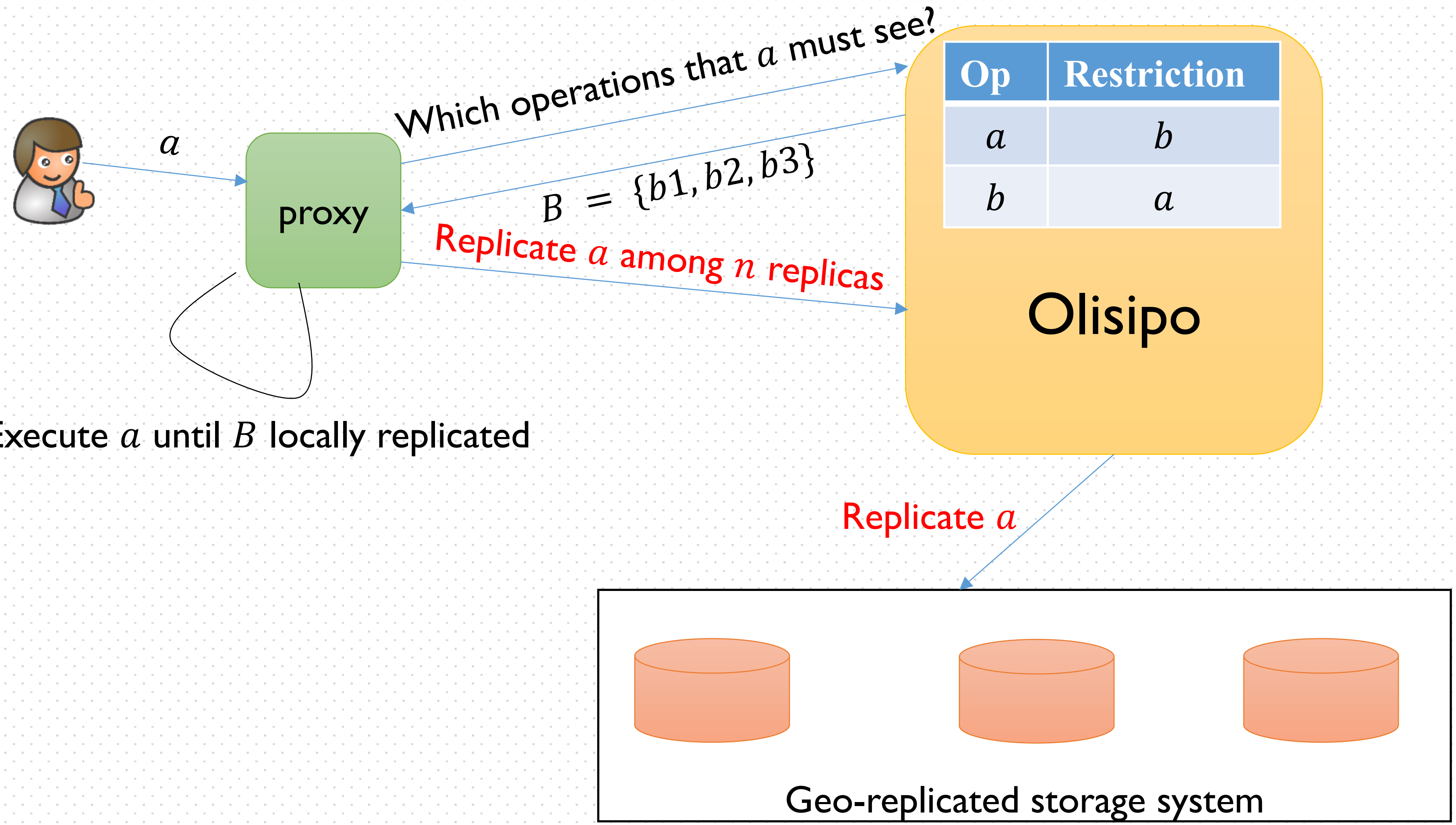
Op	Restriction
a	b
b	a

Olisipo - Overview

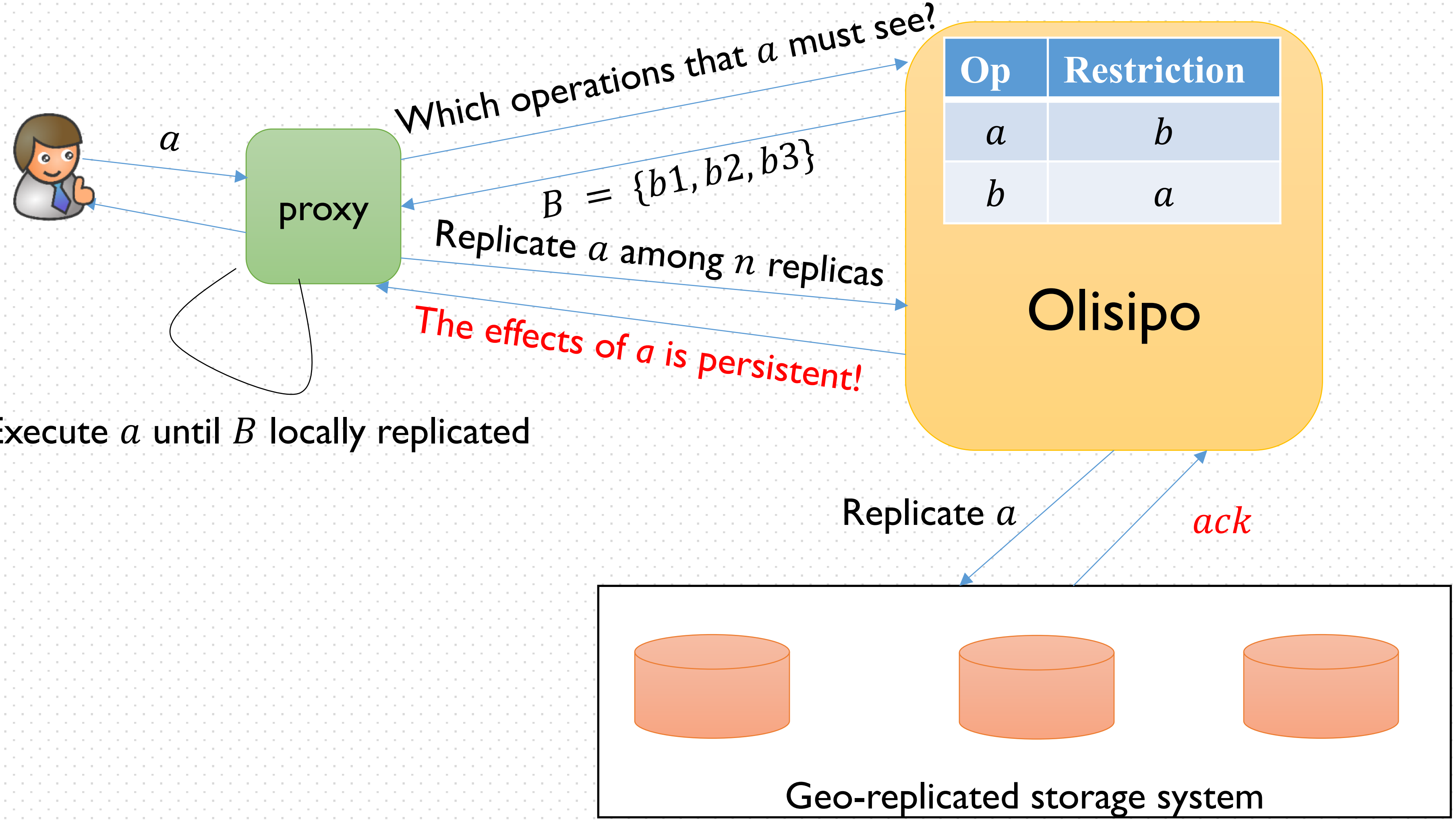


Op	Restriction
a	b
b	a

Olisipo - Overview



Olisipo - Overview



Op	Restriction
a	b
b	a

Outline

USTC, CHINA ADSLAB

1. Background and problem statement
2. Partial-Order Restrictions (PoR) Consistency
3. Olisipo: PoR consistent coordination service
4. **Evaluation and results**
5. **Conclusion**

Case study

RUBiS

- An e-commerce benchmark that emulates an auction site
- 3 invariants corresponding to 3 I-conflict sets
  - *{registerUser', registerUser'}*
  - *{storeBuyNow', storeBuyNow'}*
  - *{placeBid', closeAuction'}*

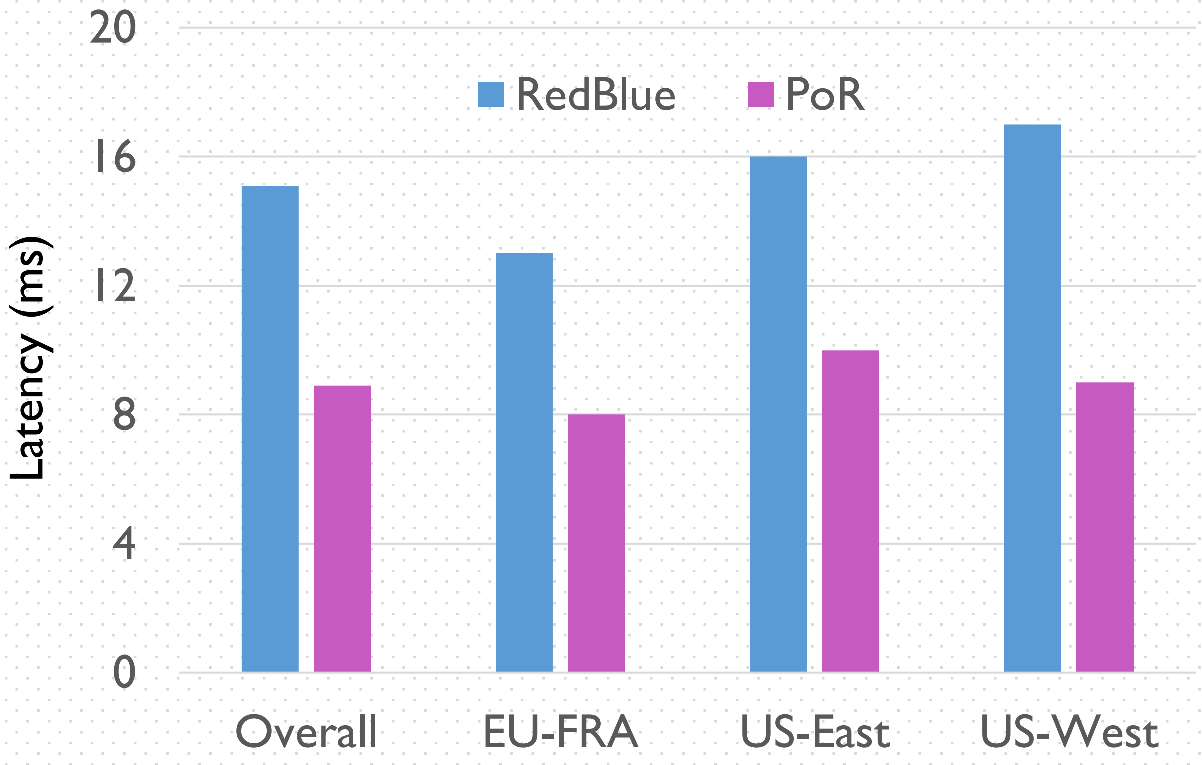
RedBlue consistency	PoR consistency
10 restrictions	*3 restrictions*

PoR consistency places fewer restrictions than RedBlue!

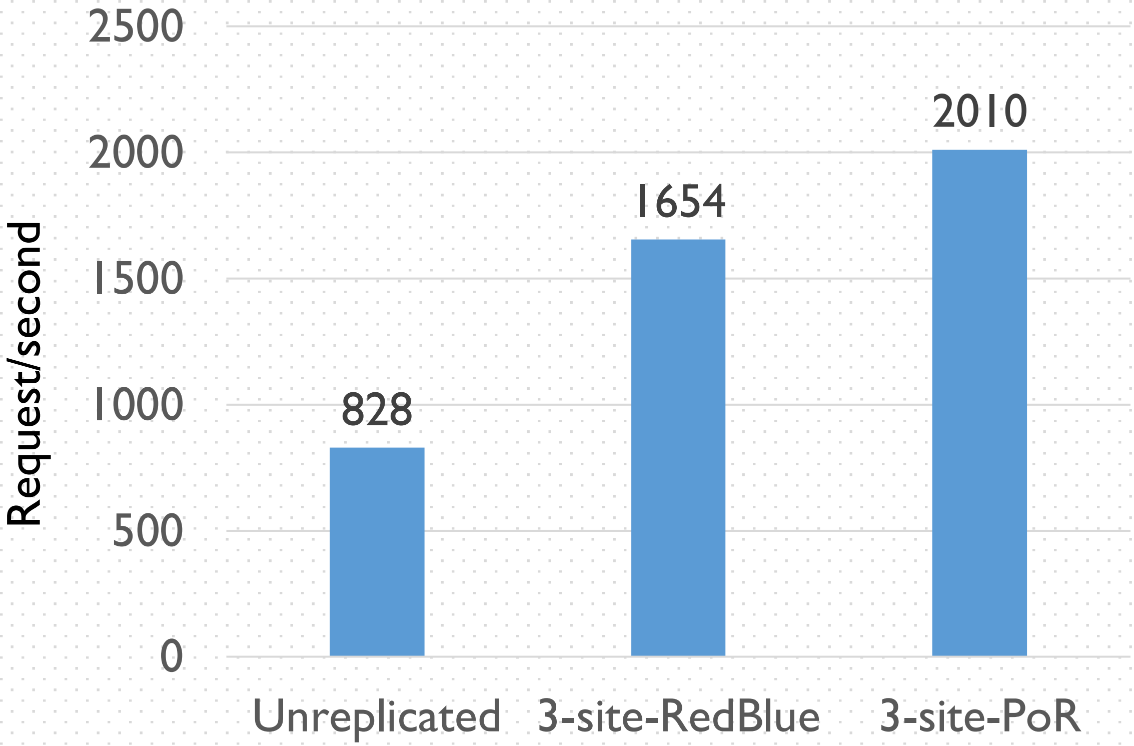
Experimental setup

- Replicating RUBiS across three regions in EC2 platform
  - EU-FRA, US-EAST, US-WEST
- Baselines:
  - Unreplicated RUBiS offering strong consistency
  - Three-region RUBiS replication under RedBlue consistency
- Questions to answer:
  - User observed latency improvement
  - Peak throughput improvement
  - Performance impact when choosing different coordination policy

Latency and throughput improvement

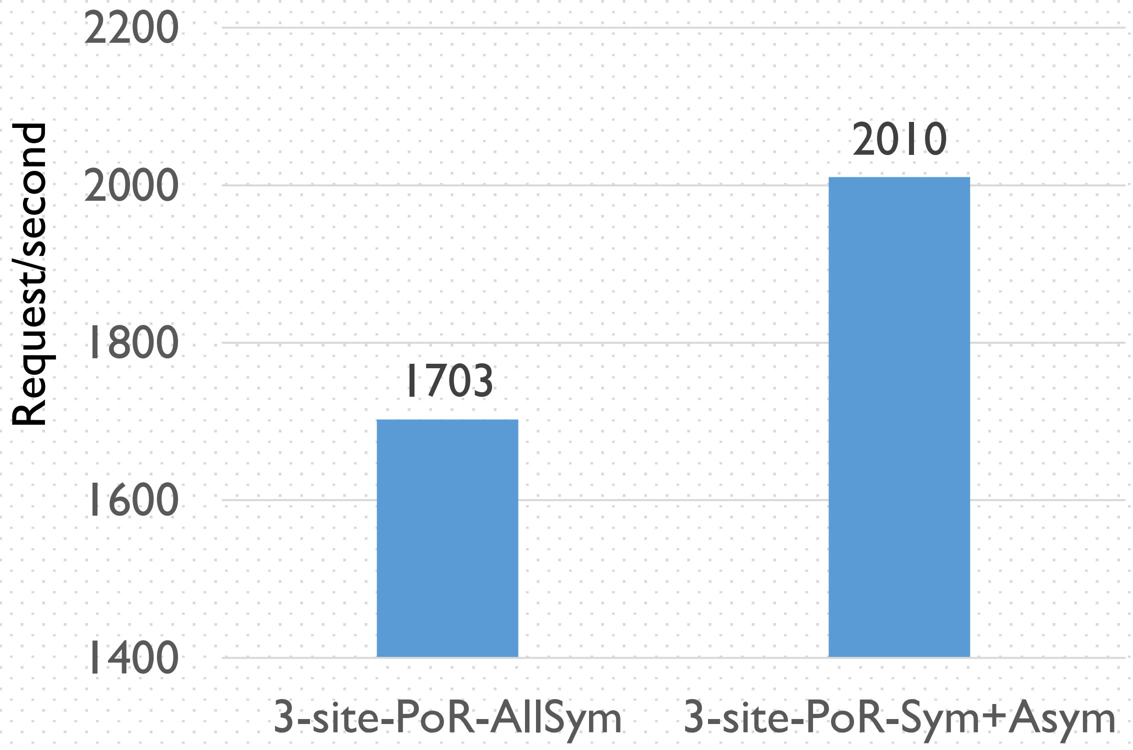


User observed latency
(lower is better)

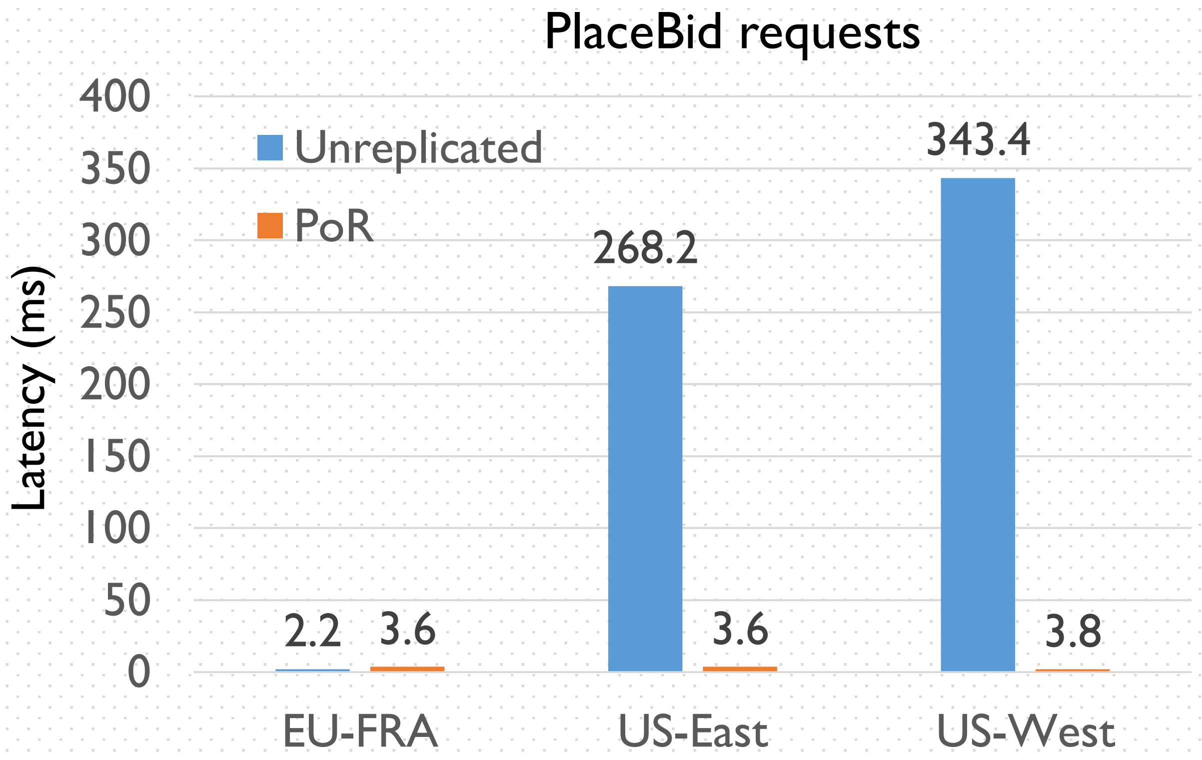


Peak throughput
(higher is better)

Choosing different coordination policies



Improper choice leads to performance penalty



Proper choice makes latency for requests demanding coordination as local access

Outline

USTC, CHINA ADSLAB

Conclusion

- Fundamental tension between performance and consistency
- PoR consistency maps consistency semantics to a minimal set of visibility restrictions over a pair of operations.
- Olisipo enforces all restrictions throughout all executions of a geo-replicated system.
- Results show that PoR consistency places fewer restrictions and achieves better performance than RedBlue consistency.

Fine-grained consistency for geo-replicated systems

Cheng Li, Nuno Preguica, Rodrigo Rodrigues

Thanks for your attention!