NOVALINCS

PUTTING CONSISTENCY BACK INTO EVENTUAL CONSISTENCY

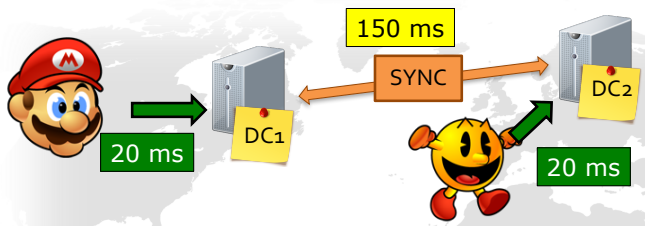
Valter Balegas, Sérgio Duarte, Carla Ferreira, Rodrigo Rodrigues, Nuno Preguiça
NOVA LINCS / U. Nova de Lisboa

Mahsa Najafzadeh, Marc Shapiro
INRIA, LIP6

INTERNET SERVICES NOWADAYS

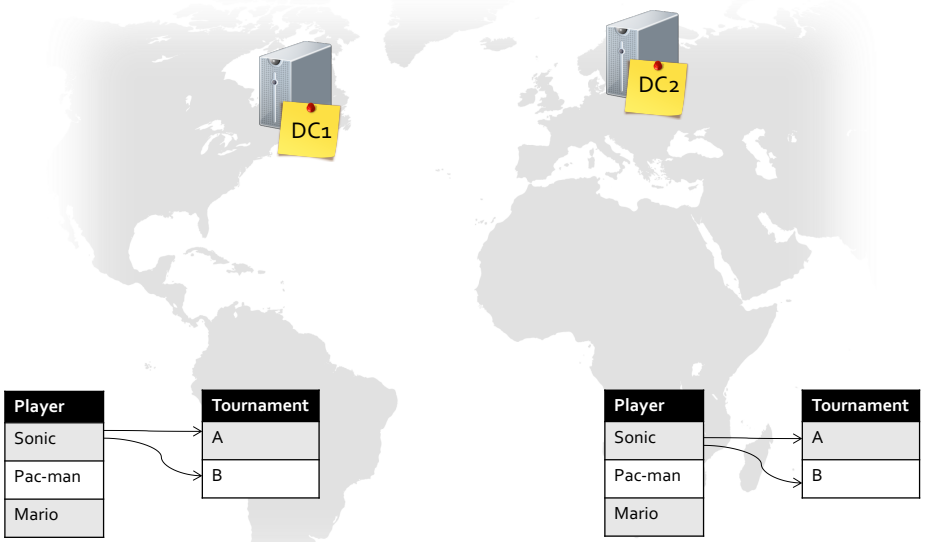
- Services operate on a global scale.
- An unprecedented number of people are using internet services.

GEO-REPLICATION

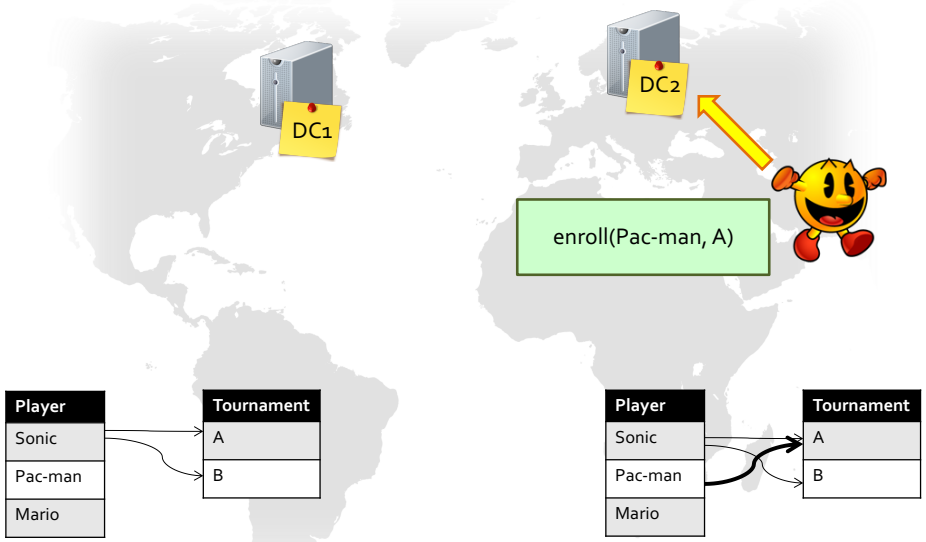


GEO-REPLICATION

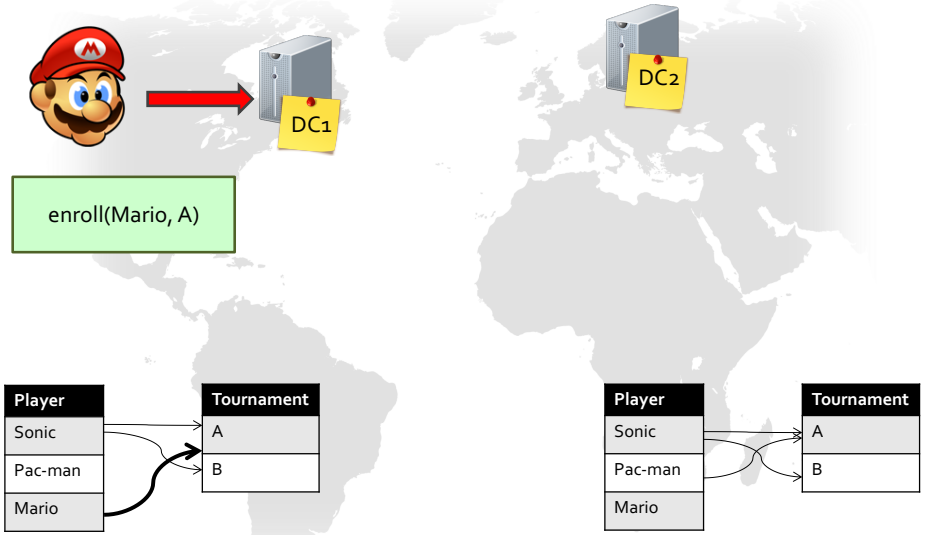
GEO-REPLICATION



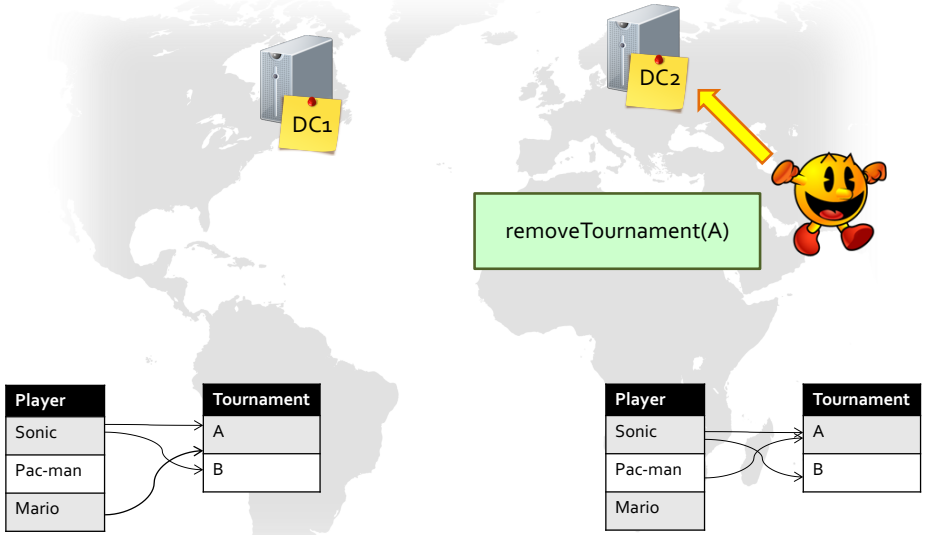
GEO-REPLICATION



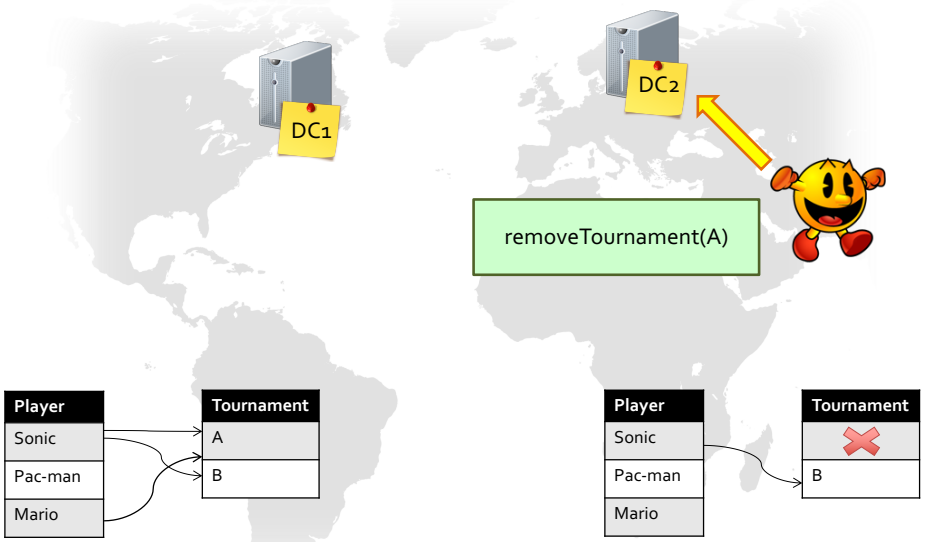
GEO-REPLICATION



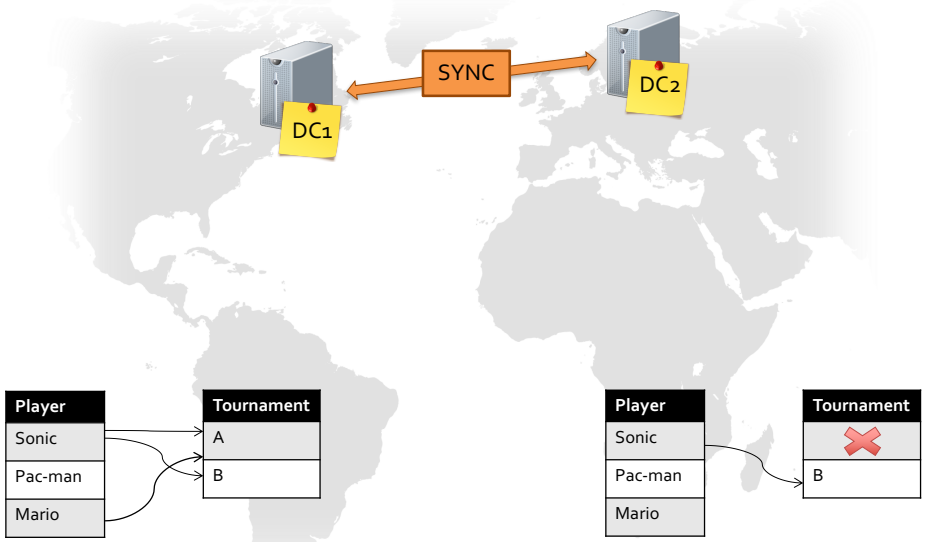
GEO-REPLICATION



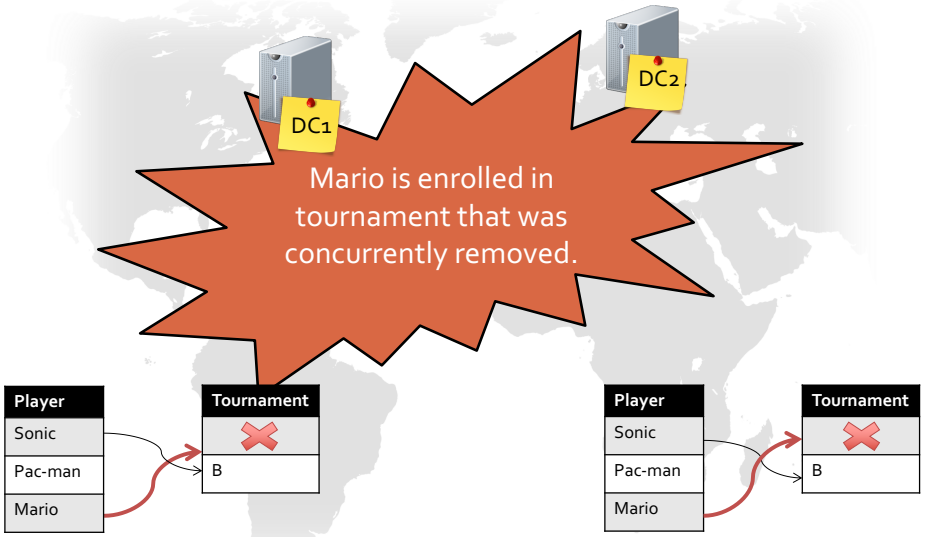
GEO-REPLICATION



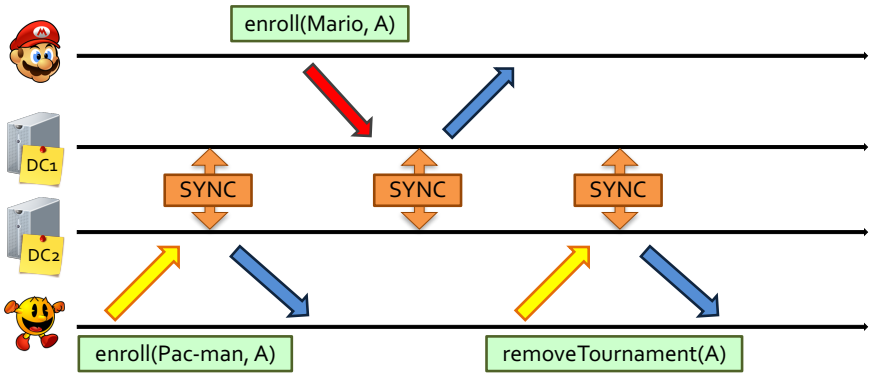
GEO-REPLICATION



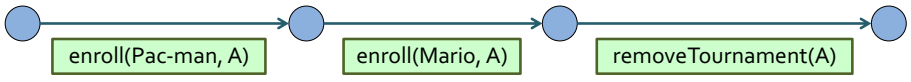
GEO-REPLICATION



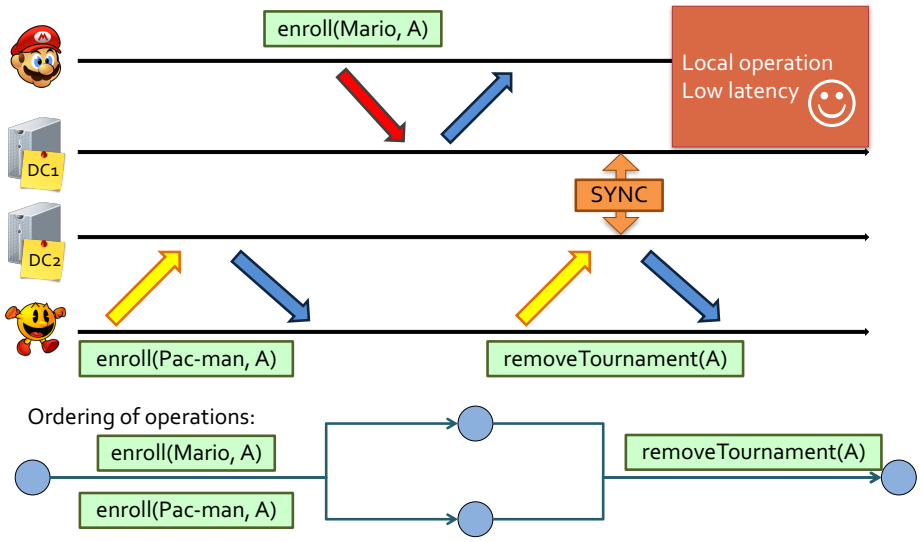
STRONG CONSISTENCY



Ordering of operations:



IS COORDINATION NEEDED?



OUTLINE

- Background
- Explicit Consistency
- Indigo
- Evaluation
- Conclusion

EXPLICIT CONSISTENCY

- Programmer specifies application invariant.
- System ensures that every state transition preserves the invariant.
- Opportunity to improve performance by not restricting execution ordering.

A METHODOLOGY FOR EXPLICIT CONSISTENCY

- Identify *I-offenders*
  - Static analysis identifies operations that may break invariants when executed concurrently.
- Choose reservations
  - Efficient mechanism to execute *I-offenders* avoiding coordination.
- Instrument application code with selected mechanism.

A METHODOLOGY FOR EXPLICIT CONSISTENCY

- Identify *I-offenders*
  - Static analysis identifies operations that may break invariants when executed concurrently.
- Choose reservations
  - Efficient mechanism to execute *I-offenders* avoiding coordination.
- Instrument application code with selected mechanism.

STATIC ANALYSIS: APPLICATION MODEL

- Programmer specifies:
  - Invariant:

    "Players can only participate in existing tournaments."

    $Inv = enrolled(p,t) \Rightarrow player(p) \wedge tournament(t)$

  - Operations' side effects:

    enroll (p,t): {*enrolled*(p,t) := true }

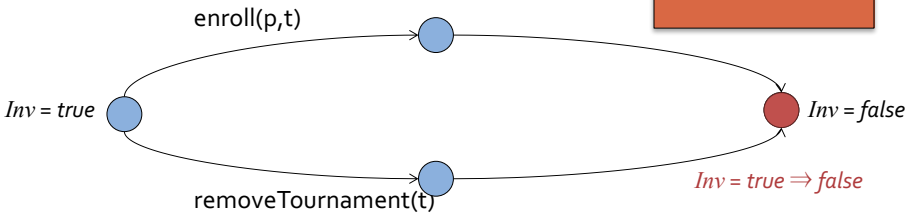
    removeTournament(t): { *tournament*(t) := false}

STATIC ANALYSIS: ALGORITHM

$Inv = enrolled(p,t) \Rightarrow player(p) \wedge tour$

$Inv = true \Rightarrow player(p) \wedge tournament(t)$

Use SMT Solver to test all pairs of operations.



$Inv = enrolled(p,t) \Rightarrow false$

A METHODOLOGY FOR EXPLICIT CONSISTENCY

- Identify *I-offenders*
  - Static analysis identifies operations that may break invariants when executed concurrently.
- Choose reservations
  - Efficient mechanism to execute *I-offenders* avoiding coordination.
- Instrument application code with selected mechanism.

RESERVATIONS

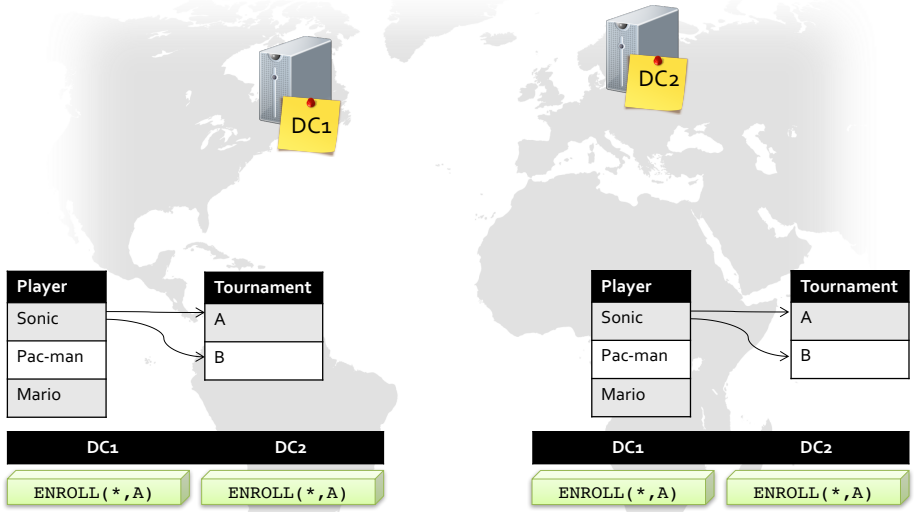
- Mechanisms to control the execution of *I-offenders* without breaking invariants.
- Coordination outside the operation flow.
- Different reservations for different invariants:

Invariant type	Reservation
Generic	Multi-level Lock
Numeric	Escrow
Referential Integrity	Multi-level Lock
Uniqueness	UID Generator
Disjunction	Multi-level Mask
Range partition	Partition Lock

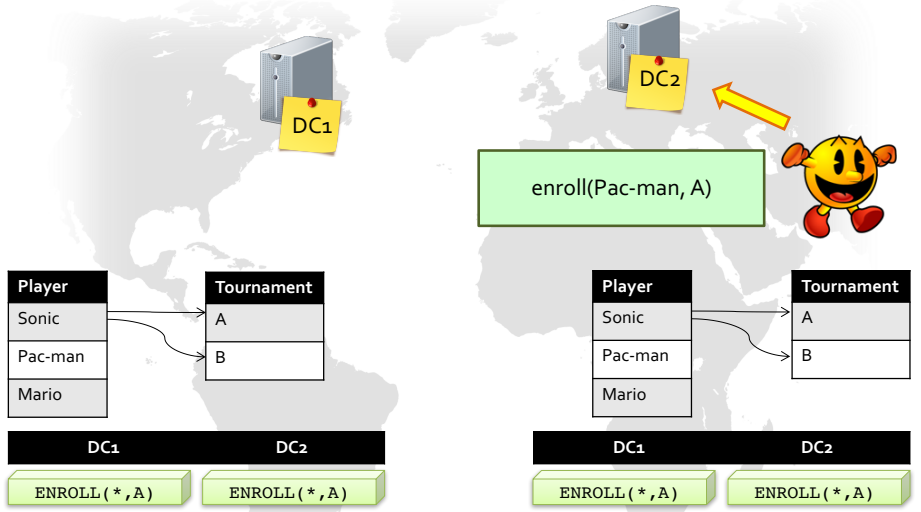
RESERVATIONS: MULTI-LEVEL LOCK

- Protects the execution of conflicting operations.
- Only allow the execution of one type of operation at a time.
- Operation can be executed by multiple clients that hold the lock.

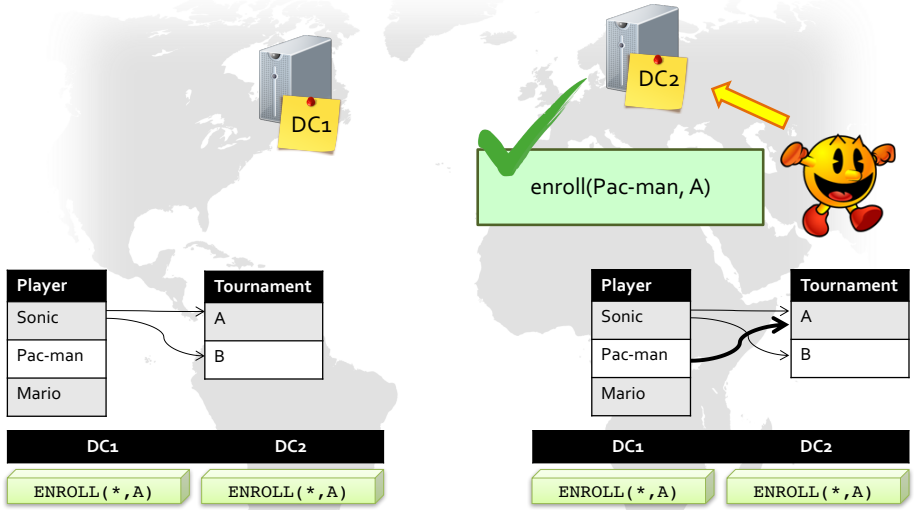
RESERVATIONS: EXAMPLE



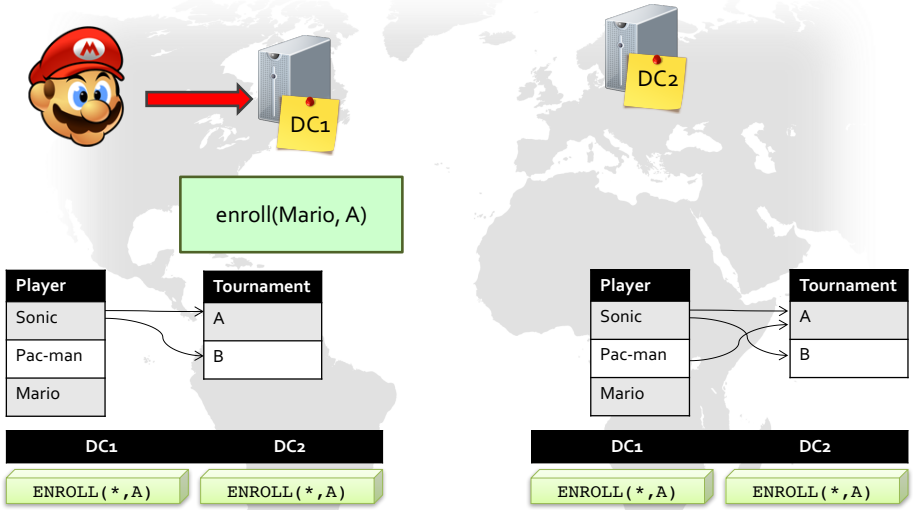
RESERVATIONS: EXAMPLE



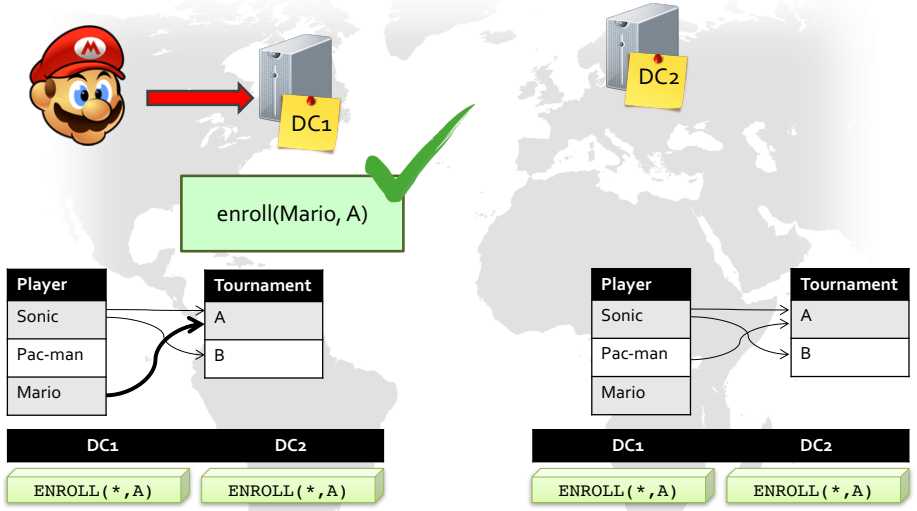
RESERVATIONS: EXAMPLE



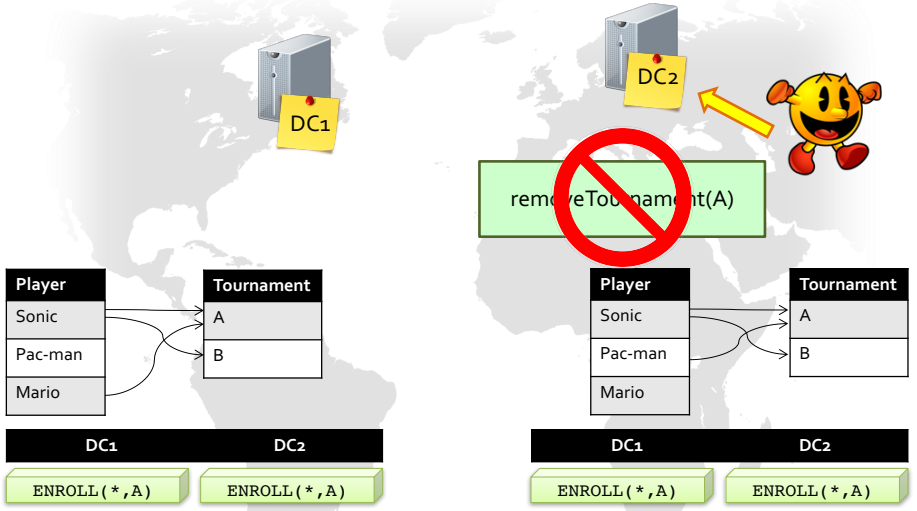
RESERVATIONS: EXAMPLE



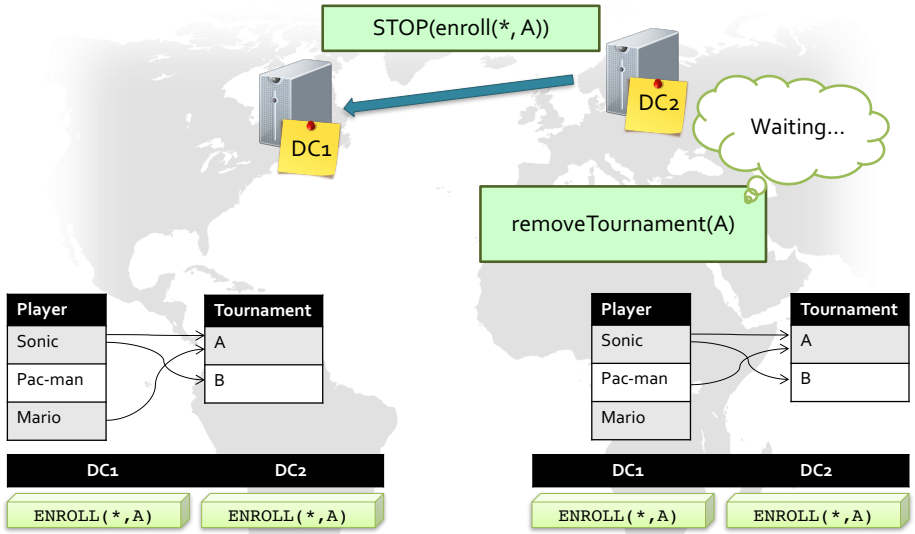
RESERVATIONS: EXAMPLE



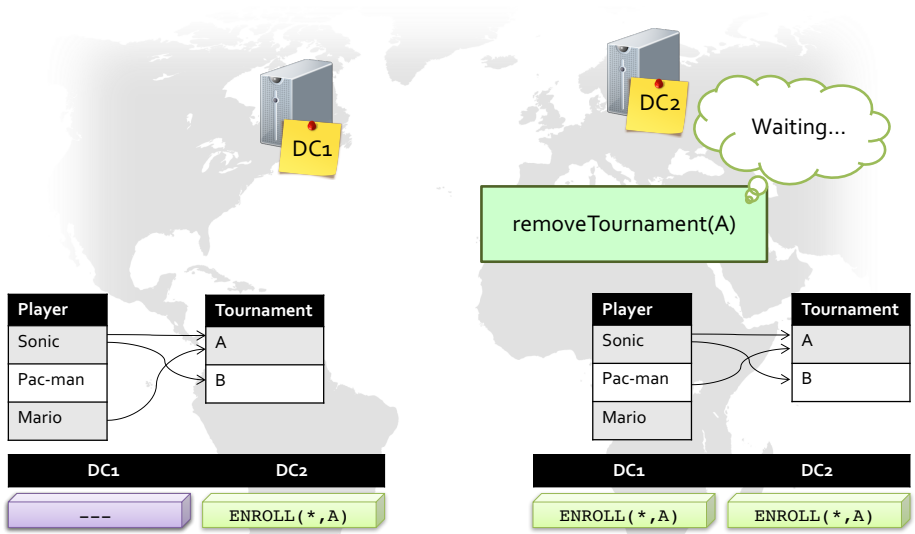
RESERVATIONS: EXAMPLE



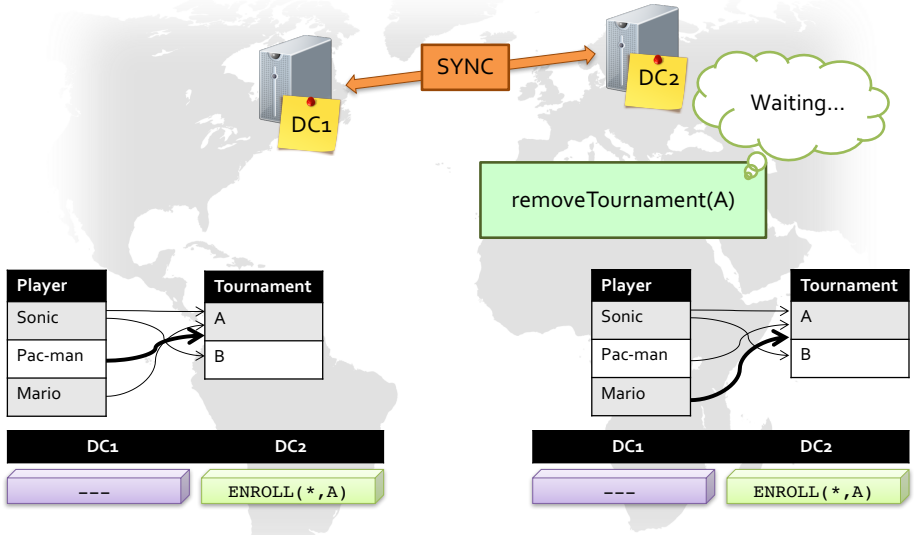
RESERVATIONS: EXAMPLE



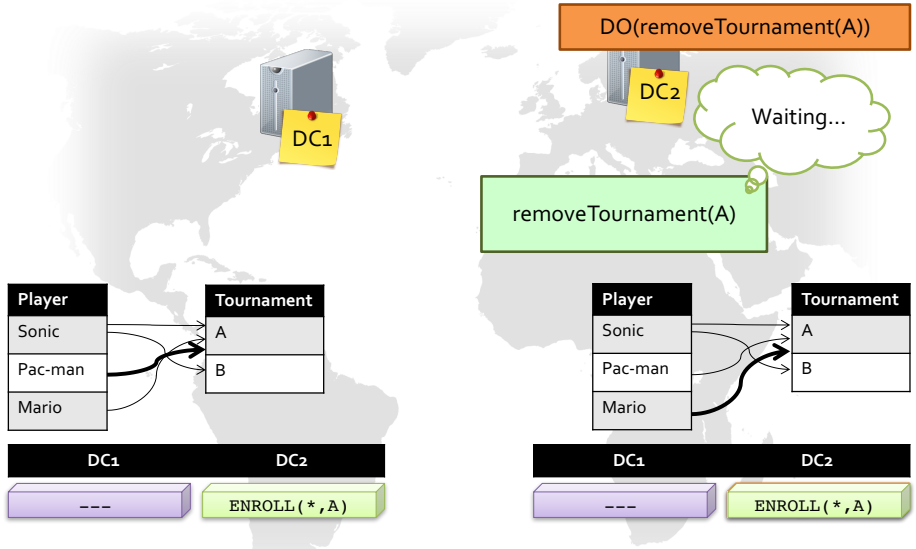
RESERVATIONS: EXAMPLE



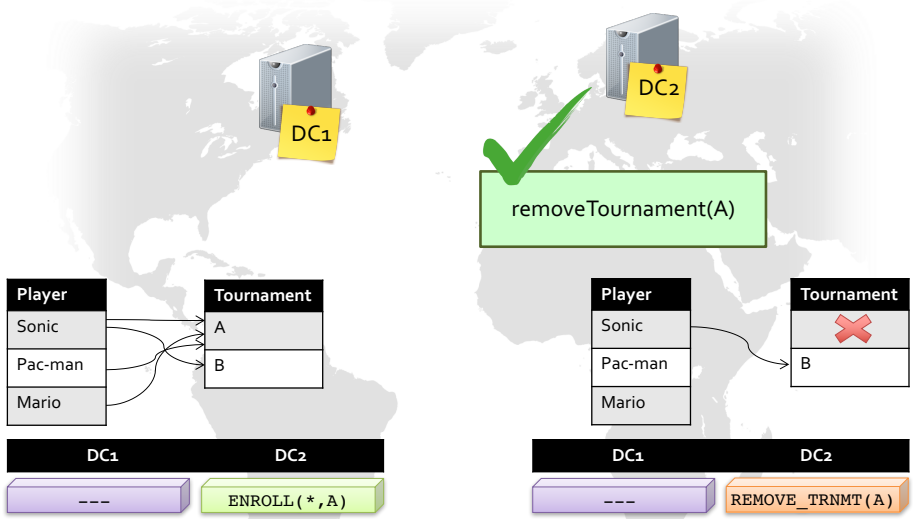
RESERVATIONS: EXAMPLE



RESERVATIONS: EXAMPLE



RESERVATIONS: EXAMPLE



INDIGO

- Middleware that provides Explicit consistency on top of KV-Stores.
- Requires only properties that are known to be efficient.
- Can be extended with new reservations.

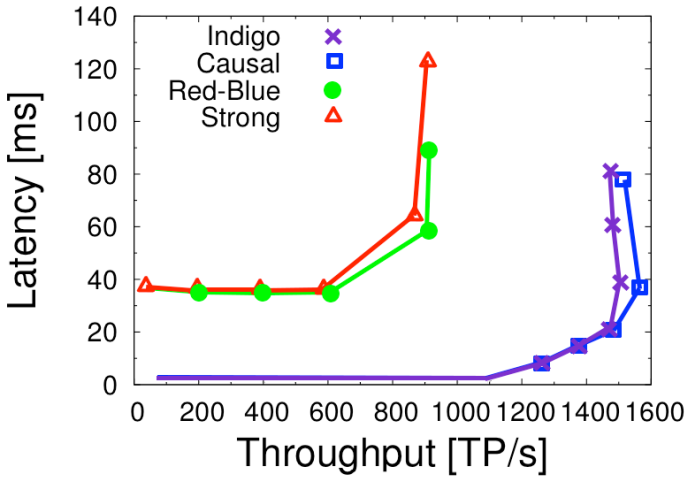
EVALUATION

- How well does the system scale?
- What is the latency of operations?
- Behavior with more reservations per operation?
- Applicability of the solution.

DEPLOYMENT

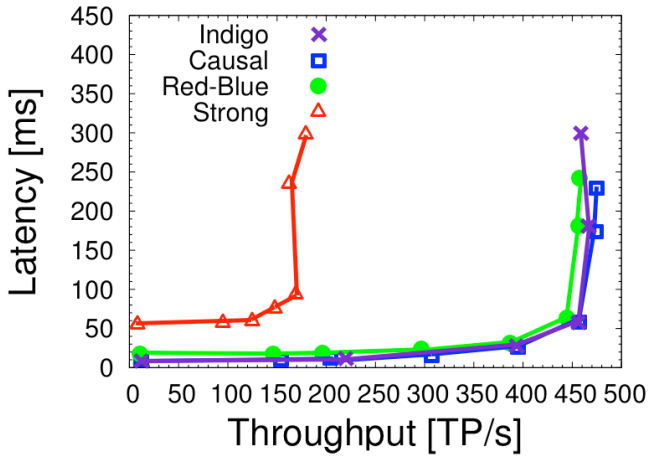
- Data-centers deployed in AWS:
  - 3 Regions (EU, US-EAST/WEST);
  - N app-servers connect to local DBs;
  - Clients submit operations to the app-server in close loop.
- Compare performance:
  - Causal Consistency
  - Strong Consistency (Writes to single server)
  - Red-Blue Consistency (Causal + Writes to single server)
  - Explicit Consistency (Causal + Reservations)

AD-SERVICE



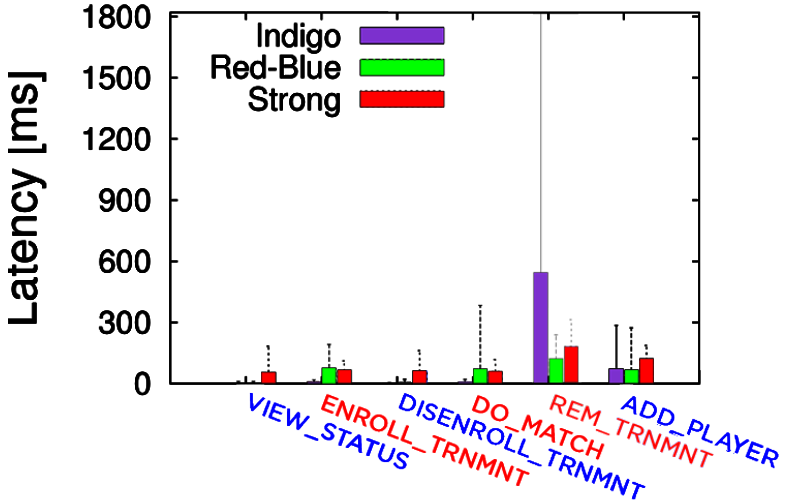
1000 ads; 100% I-offenders

TOURNAMENT



82% reads; 4% safe writes; 14% I-offenders

TOURNAMENT: OPERATIONS LATENCY



Detailed Operations Latency

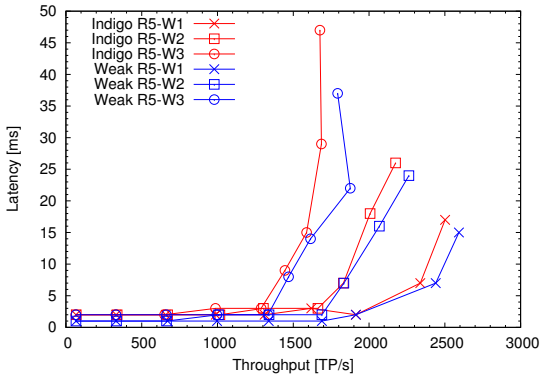
CONCLUSIONS

- Explicit Consistency successfully reduces coordination:
  - Programmers provide simple annotations;
  - Static analysis detects conflicting operations;
  - Low-latency operations with reservations.

- Performance comparable to Causal consistency.

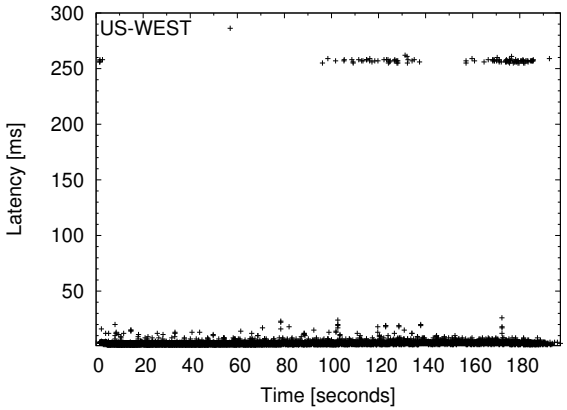
QUESTIONS?

Adding more reservations



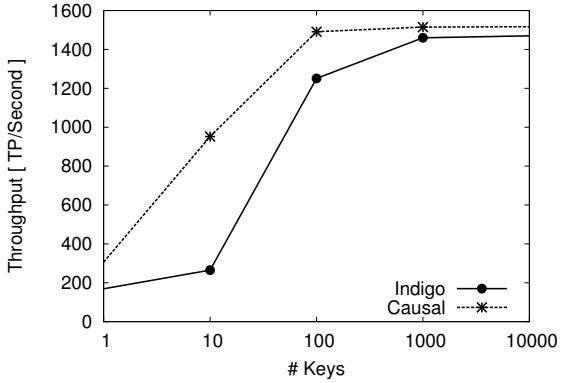
(b) Peak throughput with an increasing number of invariants (ad counter application).

Latency over time



(c) Latency of individual operations of US-W datacenter (ad counter application).

Overhead with increasing contention



(a) Peak throughput with increasing contention (ad counter application).